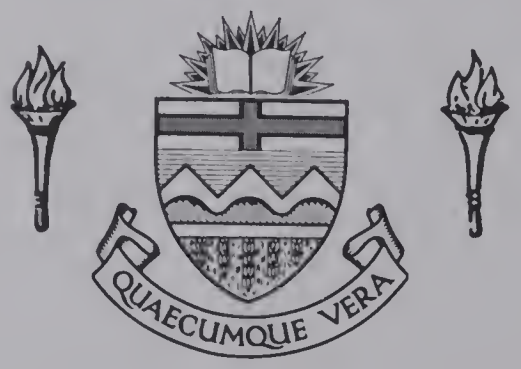


The University of Alberta
Printing Department
Edmonton, Alberta

THE UNIVERSITY OF ALBERTA

ERROR ANALYSES OF TWO ELIMINATION METHODS OF COMPUTING GENERALIZED INVERSES

by

D. Dale Olesky

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

April, 1968

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled ERROR ANALYSES OF TWO ELIMINATION METHODS OF COMPUTING GENERALIZED INVERSES submitted by D. Dale Olesky in partial fulfilment of the requirements for the degree of Master of Science.

ABSTRACT

This thesis reviews two algorithms, one for the computation of a Reflexive Generalized Inverse and the other for the computation of the Pseudoinverse of an arbitrary matrix. Upper bounds are obtained for the round-off error incurred in performing these computations. Several Reflexive Generalized Inverses and Pseudoinverses are computed, and a comparison is made between the predicted bounds and the actual round-off errors.

ACKNOWLEDGEMENTS

I express my appreciation to Professor S. Charmonman for the guidance given me in the preparation of this thesis, to Professor U.M. von Maydell for her interest and assistance in this topic, and to Professor D.B. Scott, Head of the Department of Computing Science, for providing computing facilities and financial assistance while this research was being done. I also wish to thank the National Research Council of Canada for financial assistance for carrying out this research.

TABLE OF CONTENTS

Page

CHAPTER I - INTRODUCTION

1.1 History 1
1.2 Purpose of the Study 4

CHAPTER II - BASIC THEORY

2.1 Matrix Algebra 5
2.1.1 Notation and Basic Definitions 5
2.1.2 Permutation Matrices 6
2.1.3 Hermite Normal Form 7
2.1.4 Matrix Norms 9

2.2 The Elimination Methods of Gauss and Jordan 10
2.2.1 Gaussian Elimination 10
2.2.1.1 Partial Pivoting 12
2.2.1.2 Complete Pivoting 15
2.2.2 Jordan Elimination 16

2.3 Error Analysis 21
2.3.1 Basic Operations 22
2.3.2 Inner Product Computations 24
2.3.3 Matrix Operations 32

CHAPTER III - A NORM BOUND ON THE ERROR IN COMPUTING A REFLEXIVE GENERALIZED INVERSE

3.1 Computation of a Reflexive Generalized Inverse 35

3.2 An Error Analysis 40
3.2.1 A Bound on $\|F_1\|_\infty$ 42
3.2.2 A Smaller Bound on $\|F_1\|_\infty$ 52
3.2.3 A Bound on $\|F_2\|_\infty$ 57
3.2.4 A Bound on $\|F_3\|_\infty$ 62
3.2.5 A Bound for the Computed Reflexive Generalized Inverse 65

Page

CHAPTER IV - A NORM BOUND ON THE ERROR IN COMPUTING THE PSEUDOINVERSE

4.1 Computation of the Pseudoinverse 67

4.2 An Error Analysis 70
4.2.1 A Bound for the Computed Hermite Normal Form 70
4.2.2 A Bound for the Product P*AB* 77
4.2.3 A Bound in Computing $(P^*AB^*)^{-1}$ 82
4.2.4 A Bound for the Computed Pseudoinverse 87

CHAPTER V - NUMERICAL RESULTS

5.1 Test Data 90
5.1.1 Nonsingular Matrices 90
5.1.2 Rectangular Matrices with Known Reflexive Generalized Inverses 94
5.1.3 Rectangular Matrices with Known Pseudoinverses 96

5.2 Summary of the Results 99
5.2.1 Results for the Reflexive Generalized Inverse 100
5.2.2 Results for the Pseudoinverse 106

CHAPTER VI - CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

6.1 Conclusions 112

6.2 Suggestions for Future Research 113

BIBLIOGRAPHY 116

APPENDIX - LISTINGS OF THE FORTRAN IV SUBPROGRAMS 120

LIST OF TABLES

		Page
5.1	Error in the Reflexive Generalized Inverse	101
5.2	Error in the Pseudoinverse	107

CHAPTER I

INTRODUCTION

1.1 History

The concept of the inverse of a nonsingular matrix was first generalized to include all matrices by Moore [11]. The generalized inverse, or "general reciprocal", as Moore called it, was independently rediscovered by Penrose [13] in 1955. Since then, many persons have investigated its properties. Rado [15] showed the equivalence of the definitions given by Moore and Penrose. Other definitions have been given by Rao [16] and by Goldman and Zelen [4]. Some practical methods of computation for the different types of generalized inverses have recently been suggested by Rao [16], Pyle [14], Ben-Israel and Cohen [1], Graybill, Meyer and Painter [5], Rust, Burrus and Schneeberger [18], and Willner [23].

Following Greville [6] and Rohde [17], we will call the generalized inverse as defined by Moore and Penrose a Pseudoinverse. It has often been called the Moore-Penrose Generalized Inverse or simply the Generalized Inverse. The Pseudoinverse is defined by the following theorem due to Penrose [13]. We denote by A^* the conjugate transpose of a matrix A.

Theorem 1.1 *The four equations*

(1.1) $$AXA = A,$$

(1.2) $$XAX = X,$$

(1.3) $$(XA)^* = XA$$

and

(1.4) $$(AX)^* = AX$$

have a unique solution for any matrix A.

The unique solution of equations (1.1), (1.2), (1.3), and (1.4) is the Pseudoinverse of A and is written as A^+. If A is of dimension m-by-n, then A^+ will be of dimension n-by-m.

Goldman and Zelen [4] have proposed the following generalized inverse.

Definition 1.1 *Let* A *be any real matrix of dimension m-by-n. Then a Weak Generalized Inverse of* A *is a matrix* $A^{(n)}$ *of dimension n-by-m satisfying equations (1.1), (1.2) and (1.3), i.e.*

(1.5) $$A\,A^{(n)}\,A = A,$$

(1.6) $$A^{(n)}\,A\,A^{(n)} = A^{(n)}$$

and

(1.7) $$(A^{(n)}A)' = A^{(n)}\,A,$$

where the superscript prime in (1.7) denotes matrix transposition.

A proof of the existence of $A^{(n)}$ is furnished by Goldman and Zelen. The solution of (1.5), (1.6) and (1.7), however, is not unique. The notation $A^{(n)}$ for the Weak Generalized Inverse is due to Rohde [17], who called it the Normalized Generalized Inverse.

Rao [16] has shown the existence of another non-unique generalized inverse, for which Rohde [17] states the following definition.

Definition 1.2 *A solution G to the equations*

$$AGA = A \tag{1.8}$$

and

$$GAG = G \tag{1.9}$$

is called a Reflexive Generalized Inverse of a matrix A.

Rao's proof for the existence of a Reflexive Generalized Inverse consists of a constructive procedure which may be used to compute a Reflexive Generalized Inverse (see Chapter III).

A Generalized Inverse $A^{(g)}$ of an arbitrary matrix A satisfying only equation (1.1) has been proposed by Rao [16]. Rohde [17] states the following definition:

Definition 1.3 *A matrix* $A^{(g)}$ *is said to be a Generalized Inverse of the matrix* A *if* $A\ A^{(g)}\ A = A$.

Rao originally defined $A^{(g)}$ to be a matrix such that for any vector y for which $Ax = y$ is consistent, $x = A^{(g)}y$ is a solution. As a result of this definition, Rao proves that $A^{(g)}$ is a Generalized Inverse of A if and only if $A\ A^{(g)}\ A = A$. The matrix $A^{(g)}$ is also not unique.

1.2 Purpose of the Study

Wilkinson [21] has stimulated interest in the study of the cumulative effect of round-off errors when performing computations on a digital computer. An important criterion in the choice of a suitable method for the computation of a particular type of generalized inverse is the magnitude of the upper bound for the round-off error incurred in the computation. In this paper we attempt to find upper bounds for the round-off error incurred in the computation of the Pseudoinverse and a Reflexive Generalized Inverse. The computational procedures due to Willner [23] and Rao [16], respectively, are examined.

CHAPTER II

BASIC THEORY

2.1 Matrix Algebra

In this section we discuss some of the basic properties of matrices which are used throughout this paper. Some particular types of matrices, which appear in the computation of generalized inverses, are defined. The concept of the norm of a matrix is introduced, and various types of matrix norms, including the norm of a rectangular matrix, are also considered.

2.1.1 Notation and Basic Definitions Upper-case Latin letters and upper-case Greek letters designate matrices, and all lower-case letters designate scalars. We say that a rectangular matrix $A = (a_{ij})$ with m rows and n columns is of dimension m-by-n. If $m = n$, the matrix A is square and of order n. The zero matrix is denoted by O and the identity matrix by I. The complex conjugate of a matrix A is denoted by A^*. If A is a real matrix, A^* is written as A', the transpose of A. A^{-1} designates the inverse of a nonsingular matrix A. The matrix whose elements are the absolute values of the corresponding elements of A is denoted by $|A|$, i.e., $|A| = (|a_{ij}|)$.

For a square matrix A of order n, the elements a_{ii}, $i=1,2,\ldots,n$, constitute the main diagonal of A. For

$$a_{ij} = 0 , \quad i \neq j ,$$

and at least one nonzero element of the main diagonal, A is said to be a diagonal matrix. Moreover, for

$$a_{ij} = 0 , \quad i < j ,$$

A is called a lower-triangular matrix, and for

$$a_{ij} = 0 , \quad i > j ,$$

an upper-triangular matrix.

2.1.2 Permutation Matrices According to Wilkinson [22], permutation matrices may be defined as follows:

Definition 2.1 *A matrix* $P_{\alpha_1,\alpha_2,\ldots,\alpha_n}$ *of order* n *is a permutation matrix if*

$$p_{i,\alpha_i} = 1$$

and

$$p_{ij} = 0 , \quad j \neq \alpha_i ,$$

where $(\alpha_1,\alpha_2,\ldots,\alpha_n)$ *is some permutation of* $(1,2,\ldots,n)$.

These matrices have the property that one and only one element of every row and column is nonzero and equal to 1.

The permutation matrices used in this paper are a subset of the set of all permutation matrices as defined above. They are obtained by interchanging two rows of the identity matrix of order n. We let R_{ij} designate the permutation matrix obtained by interchanging rows i and j of I.

Pre-multiplication of an arbitrary matrix A of dimension m-by-n by a permutation matrix R_{ij} has the effect of interchanging rows i and j of A. Post-multiplication of an arbitrary matrix A of dimension m-by-n by R_{ij} interchanges columns i and j of A. We will denote a permutation matrix used for post-multiplication by C_{ij} rather than R_{ij}.

2.1.3 Hermite Normal Form Before defining the Hermite normal form of a matrix, we will define what is meant by an elementary row operation.

Definition 2.2 *Three elementary row operations may be defined on a matrix A of dimension m-by-n:*

Type I: *the interchange of two rows of A;*

Type II: *the addition of a scalar multiple of one row of A to another row of A;*

Type III: the multiplication of a row of A by a nonzero scalar.

The definition of the Hermite normal form of a matrix and the computational procedure are due to Marcus and Minc [10].

Definition 2.3 *Let A be a matrix of dimension m-by-n with rank k. Then the Hermite normal form H of A is a matrix of dimension m-by-n such that:*

(i) the first k rows of H are nonzero and the remaining m - k rows are zero;

(ii) the first nonzero element in the i-th row of H, i=1,2,...,k, is 1 and occurs in column n_i, where $n_1 < n_2 < \ldots < n_k$; and,

(iii) the only nonzero element in column n_i of H is the 1 in the i-th row.

Using elementary row operations, the reduction of A to H can be performed by repeating the following two steps for i=1,2,...,k.

(i) Let n_i denote the column number of the i-th column of A with a nonzero entry in rows i, i+1,...,m. By a Type I row operation bring a nonzero entry to the (i,n_i) position from rows i,i+1,...,m. Then make this element equal to 1 by a Type III row operation.

(ii) With Type II row operations, annihilate all of the elements of column n_i except the element (i,n_i).

The computation of the Hermite normal form is essential in Willner's algorithm for the computation of the Pseudo-inverse. We have programmed the above procedure, incorporating a type of partial pivoting (see Section 2.2).

2.1.4 Matrix Norms A norm of a matrix is a non-negative scalar which is a measure of the magnitude of the matrix. A norm of a square matrix A, denoted by $\|A\|$, must satisfy the following four conditions:

(i) $\|A\| > 0$ if $A \neq 0$ and $\|0\| = 0$;

(ii) $\|kA\| = |k|\|A\|$ for any scalar k ;

(iii) $\|A+B\| \leq \|A\| + \|B\|$;

(iv) $\|AB\| \leq \|A\|\|B\|$.

Four common matrix norms defined on square matrices are:

$$\|A\|_1 = \max_j \sum_i |a_{ij}| \quad , \tag{2.1}$$

$$\|A\|_\infty = \max_i \sum_j |a_{ij}| \quad , \tag{2.2}$$

$$\|A\|_2 = (\text{maximum eigenvalue of } A^*A)^{1/2} , \tag{2.3}$$

and

$$\|A\|_E = \Big(\sum_i \sum_j |a_{ij}|^2 \Big)^{1/2} . \tag{2.4}$$

Korganoff and Pavel-Parvu [9] show that these definitions may be extended to include rectangular matrices. They mention, however, as does Householder [8], that rectangular matrices can be normed by adjoining zero rows or columns to them and then applying the definitions of norms of square matrices. This is how we shall compute the norm of a rectangular matrix.

2.2 The Elimination Methods of Gauss and Jordan

There are various well-known elimination procedures associated with the name of Gauss for solving a system of linear equations with a nonsingular coefficient matrix and for inverting nonsingular matrices. In this section we consider two of these elimination methods, Gaussian elimination and Jordan (or Gauss-Jordan) elimination. We examine, in particular, the matrix equivalents of these two elimination methods (see Fox [3]).

2.2.1 Gaussian Elimination Let A be a matrix of dimension m-by-n, where $m \leq n$, and let the rank of A be k, $k \leq m$. Then A can be reduced to a matrix of the following form (if the first k columns of A are linearly independent)

(2.5) $$A^{(k)} = \begin{bmatrix} a_{11}^{(k)} & a_{12}^{(k)} & a_{13}^{(k)} & \cdots & a_{1,k-1}^{(k)} & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} \\ 0 & a_{22}^{(k)} & a_{23}^{(k)} & \cdots & a_{2,k-1}^{(k)} & a_{2k}^{(k)} & \cdots & a_{2n}^{(k)} \\ 0 & 0 & a_{33}^{(k)} & \cdots & a_{3,k-1}^{(k)} & a_{3k}^{(k)} & \cdots & a_{3n}^{(k)} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \hline & & & 0 & & & & \end{bmatrix}$$

by a series of k transformations.

2.2.1.1 Partial Pivoting By partial pivoting we mean the following: at the i-th stage, $i=1,2,\ldots,k$, in the transformation of A into $A^{(k)}$, the pivotal element is chosen to be the element of largest magnitude in the i-th column and in rows $i,i+1,\ldots,m$. If, at the i-th stage, the pivotal element is in the j-th row, then rows i and j are interchanged. This interchange can be accomplished by pre-multiplication by the permutation matrix R_{ij}. We now omit the subscripts i and j of R_{ij}, and write instead R_i to indicate that this is the permutation matrix which accomplishes the partial pivoting at the i-th stage.

In terms of matrices, the partial pivoting at the first stage may be represented by the matrix product $R_1 A$. Then the first reduced matrix will be $J_1 R_1 A$, where J_1 is a lower-triangular matrix of the form

$$J_1 = \begin{bmatrix} 1 & & & & & \\ m_{21} & 1 & & & & \\ m_{31} & 0 & 1 & & & \\ \vdots & \vdots & \vdots & & & \\ m_{m1} & 0 & 0 & \ldots & 0 & 1 \end{bmatrix}, \tag{2.6}$$

where

$$m_{i1} = -\frac{a_{i1}}{a_{11}} \quad . \tag{2.7}$$

The matrix J_1R_1A will be of the form

$$J_1R_1A = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & a_{32}^{(1)} & a_{33}^{(1)} & \cdots & a_{3n}^{(1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & a_{m2}^{(1)} & a_{m3}^{(1)} & \cdots & a_{mn}^{(1)} \end{bmatrix} . \tag{2.8}$$

The reduced form of A, $A^{(k)}$, can be obtained from the following recursive relations:

$$\begin{cases} A^{(0)} = A \\ A^{(i)} = J_iR_iA^{(i-1)}, \qquad i=1,2,\ldots,k, \end{cases} \tag{2.9}$$

where each J_i is a lower-triangular matrix of the form

$$
(2.10)\qquad J_i = \begin{bmatrix}
1 & & & & & & & \\
0 & 1 & & & & & & \\
0 & 0 & & & & & & \\
\vdots & \vdots & & & & & & \\
0 & 0 & \cdots & 1 & & & & \\
0 & 0 & \cdots & 0 & 1 & & & \\
0 & 0 & \cdots & 0 & m_{i+1,i} & 1 & & \\
0 & 0 & \cdots & 0 & m_{i+2,i} & 0 & & \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \\
0 & 0 & \cdots & 0 & m_{mi} & 0 & \cdots & 1
\end{bmatrix}
$$

with

$$
(2.11)\qquad m_{ji} = -\frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}}, \qquad \begin{matrix} i=1,2,\ldots,k, \\ j=i+1,i+2,\ldots,m, \end{matrix}
$$

where $a_{ij}^{(0)} = a_{ij}$. From (2.9), the final reduced matrix is

$$
(2.12)\qquad A^{(k)} = J_k R_k J_{k-1} R_{k-1} \cdots J_1 R_1 A \ .
$$

This process of reducing A to $A^{(k)}$ is called Gaussian forward elimination with partial pivoting. As a result of

the partial pivoting, we note that

$$|m_{ji}| \leq 1 , \qquad i=1,2,\ldots,k, \quad j=i+1,i+2,\ldots,m. \tag{2.13}$$

2.2.1.2 Complete Pivoting An alternative to partial pivoting is complete pivoting, which may be described as follows: at the i-th stage, i=1,2,...,k, in the transformation of A into $A^{(k)}$, the pivotal element is chosen to be the element of largest magnitude in columns $i,i+1,\ldots,n$ and in rows $i,i+1,\ldots,m$. In order to transfer the pivotal element to position (i,i) of the matrix, one row interchange and one column interchange are necessary. If the pivotal element at the i-th stage is $a_{rs}^{(i-1)}$, then rows i and r and columns i and s must be interchanged. This can be accomplished by pre-multiplication by R_{ir} and post-multiplication by C_{is}. We shall omit the double subscript and simply write R_i and C_i.

Thus the complete pivoting at the first stage can be represented by the matrix product R_1AC_1. Then the first reduced matrix will be $J_1R_1AC_1$, where J_1 is defined by (2.6), and will be of the same form as J_1R_1A. The final reduced form of A can be obtained from the recursive relations

$$(2.14)\qquad \begin{cases} A^{(0)} = A \\ A^{(i)} = J_i R_i A^{(i-1)} C_i, \quad i=1,2,\ldots,k, \end{cases}$$

where the matrices J_i are defined by (2.10) and $a_{ij}^{(0)} = a_{ij}$. The final reduced matrix is

$$(2.15)\qquad A^{(k)} = J_k R_k J_{k-1} R_{k-1} \ldots J_1 R_1 A C_1 C_2 \ldots C_k .$$

This process of reducing A to $A^{(k)}$ is called Gaussian forward elimination with complete pivoting.

As was the case for partial pivoting, it may be noted that

$$(2.16)\qquad |m_{ji}| \le 1 , \qquad i=1,2,\ldots,k, \quad j=i+1,i+2,\ldots,m.$$

With complete pivoting, the following inequality also holds on the elements of $A^{(k)}$:

$$(2.17)\qquad |a_{ii}^{(k)}| \ge |a_{ij}^{(k)}| , \qquad i=1,2,\ldots,k, \quad j=i+1,i+2,\ldots,n.$$

2.2.2 Jordan Elimination We again let A be a matrix of dimension m-by-n with rank k, where $m \le n$ and $k \le m$. It is well-known that Jordan elimination reduces a nonsingular

matrix of order n to diagonal form in n transformations (see Fox [3]). When applied to a rectangular matrix A, k Jordan transformations reduce A to $A^{(k)}$, where

$$(2.18)\qquad A^{(k)} = \left[\begin{array}{ccccccc} a_{11}^{(k)} & 0 & \cdots & 0 & a_{1,k+1}^{(k)} & \cdots & a_{1n}^{(k)} \\ 0 & a_{22}^{(k)} & \cdots & 0 & a_{2,k+1}^{(k)} & \cdots & a_{2n}^{(k)} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{kk}^{(k)} & a_{k,k+1}^{(k)} & \cdots & a_{kn}^{(k)} \\ \hline & & & 0 & & & \end{array}\right],$$

if the first k columns of A are linearly independent.

The first Jordan transformation is identical to the first transformation of Gaussian elimination. In terms of matrices, it can be represented as the matrix product J_1A, where J_1 is defined by (2.6). All succeeding Jordan transformations differ from the corresponding Gaussian transformations. The second Jordan transformation can be represented by the matrix product J_2J_1A, where

$$J_2 = \begin{bmatrix} 1 & m_{12} & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & m_{32} & 1 & 0 & \dots & 0 \\ 0 & m_{42} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & m_{m2} & 0 & 0 & \dots & 1 \end{bmatrix} \tag{2.19}$$

with

$$m_{i2} = -\frac{a_{i2}^{(1)}}{a_{22}^{(1)}}, \qquad i=1,3,4,\dots,m, \tag{2.20}$$

and $A^{(1)} = J_1 A$. Then $J_2 J_1 A$ will be of the form

$$J_2 J_1 A = \begin{bmatrix} a_{11}^{(2)} & 0 & a_{13}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \dots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{m3}^{(2)} & \dots & a_{mn}^{(2)} \end{bmatrix}. \tag{2.21}$$

$A^{(k)}$ can be obtained from the following recursive relations:

$$(2.22)\qquad \begin{cases} A^{(0)} = A \\ A^{(i)} = J_i A^{(i-1)}, \qquad i=1,2,\ldots,k, \end{cases}$$

where each matrix J_i has the form

$$(2.23)\qquad J_i = \begin{bmatrix} 1 & 0 & \ldots & 0 & m_{1i} & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & m_{2i} & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 1 & m_{i-1,i} & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & m_{i+1,i} & 1 & \ldots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 0 & m_{mi} & 0 & \ldots & 1 \end{bmatrix}$$

with

$$(2.24)\qquad m_{ji} = -\frac{a_{ji}^{(i-1)}}{a_{ii}^{(i-1)}}, \qquad i=1,2,\ldots,k, \quad j=1,2,\ldots,m,\ j\neq i,$$

and $a_{ij}^{(0)} = a_{ij}$. The final reduced matrix is given by

$$A^{(k)} = J_k J_{k-1} \cdots J_1 A \ . \tag{2.25}$$

In practice, some form of pivoting may be necessary to insure that none of the pivotal elements $a_{ii}^{(i-1)}$ become zero. However, to achieve the form of $A^{(k)}$ in (2.18), the pivot at the i-th stage must be chosen from rows $i,i+1,\ldots,m$. Thus it follows that

$$|m_{ji}| \leq 1 \ , \qquad \begin{matrix} i=1,2,\ldots,k, \\ j=i+1,i+2,\ldots,m, \end{matrix} \tag{2.26}$$

but the magnitudes of the multipliers m_{ji}, $j=1,2,\ldots,i-1$, are not bounded.

By analogy with the method of Gaussian elimination, the matrix equivalent of Jordan elimination with partial pivoting is

$$A^{(k)} = J_k R_k J_{k-1} R_{k-1} \cdots J_1 R_1 A \ , \tag{2.27}$$

and the matrix equivalent of Jordan elimination with complete pivoting is

$$A^{(k)} = J_k R_k J_{k-1} R_{k-1} \cdots J_1 R_1 A C_1 C_2 \cdots C_k \ . \tag{2.28}$$

In both (2.27) and (2.28), however, we cannot place a bound on the magnitudes of the multipliers, m_{ji}, of J_i above the main diagonal.

2.3 Error Analysis

In this section we state some of the fundamental floating-point bounds on the round-off error incurred in performing certain mathematical computations on a digital computer. Only floating-point computation is considered, as fixed-point computation is not suitable for storing numbers of a sufficiently large range of magnitudes or for performing lengthy calculations with numbers of varying magnitudes. All of the results presented here are due to Wilkinson [21].

The number of binary digits in a, the mantissa of a floating-point number $x = 2^b(a)$, is denoted by t. Throughout this paper we assume that the number ε is the unit round-off error, i.e.

$$|\varepsilon| \leq 2^{-t} . \tag{2.29}$$

We will be concerned only with the so-called "backward error analysis". In performing a backward error analysis on a certain computation, the actual difference between a computed value and a true value is not sought. Instead, where our mathematical computation takes the form $x = f(a_1,a_2,\ldots,a_n)$,

we attempt to show that the computed value obtained for x is exactly equal to $f(a_1+e_1,a_2+e_2,\ldots,a_n+e_n)$ for some values of the e_i. The problem, then, is to place bounds on the magnitudes of the e_i. The mathematical equation which defines the computed value of x is called a computational equation, and we write

$$x \equiv f(a_1+e_1,a_2+e_2,\ldots,a_n+e_n) , \tag{2.30}$$

followed by inequalities which limit the magnitudes of the e_i.

2.3.1 Basic Operations Let $x_1 = 2^{b_1}(a_1)$ and $x_2 = 2^{b_2}(a_2)$ be two floating-point numbers, and let fl denote a floating-point arithmetic computation. Then the following theorem gives a bound to the round-off error incurred in performing a floating-point addition, subtraction, multiplication or division with a double-precision accumulator.

Theorem 2.1

$$fl(x_1 \lambda x_2) \equiv (x_1 \lambda x_2)(1+\varepsilon) ,$$

where λ *is any one of the four arithmetic operators* $+$, $-$, $\times$, *or* $\div$.

Proof: Let the exact normalized value of $x_1 \lambda x_2$ be $2^{b_3}(a_3)$. Then

$$fl(x_1 \lambda x_2) \equiv (x_1 \lambda x_2) + \varepsilon_1 ,$$

where

$$|\varepsilon_1| \leq \frac{1}{2} 2^{-t}(2^{b_3}) .$$

Since a relative-error bound of the form

$$fl(x_1 \lambda x_2) \equiv (x_1 \lambda x_2)(1+\varepsilon_2) ,$$

is required, we can write

$$(x_1 \lambda x_2)\, \varepsilon_2 = \varepsilon_1 .$$

Therefore

$$\varepsilon_2 = \frac{\varepsilon_1}{x_1 \lambda x_2} \leq \frac{\frac{1}{2} 2^{-t}(2^{b_3})}{\frac{1}{2}(2^{b_3})} ,$$

as $|a_3| \geq \frac{1}{2}$ due to normalization of floating-point numbers.

Thus

$$|\varepsilon_2| \leq 2^{-t} \quad .$$

Q.E.D.

2.3.2 Inner Product Computations Whenever it is necessary to compute an inner product, we will assume that it is done by using floating-point accumulation, which may be defined as follows. If the inner product to be calculated is $\sum_{i=1}^{n} a_i b_i$, then each product $a_i b_i$ is computed by using single-precision floating-point arithmetic and storing the 2t-digit product without rounding. Then the n 2t-digit products are added using double-precision arithmetic, and the final result is rounded to t digits. We shall indicate the computation of an inner product using floating-point accumulation by the symbol fl_2. The symbol fl_3 denotes a floating-point arithmetic computation using a single-precision accumulator.

The following lemmas are required for the proof of the next theorem (Theorem 2.2).

Lemma 2.1 *Using a single-precision (t-digit) accumulator,*

$$fl_3(x_1+x_2) \equiv (x_1+x_2)(1+\varepsilon_1) \quad ,$$

where

$$|\varepsilon_1| \leq \frac{3}{2}\, 2^{-t} .$$

Proof: When using a single-precision accumulator, the maximum amount of round-off error will be incurred when two separate rounding errors are made in performing the addition. One rounding can occur before the addition is carried out, in shifting the smaller of the two floating-point numbers to the right to align the binary points. We will denote this rounding error by η_1. A second rounding error may be incurred after the addition takes place, if the sum of the mantissas is greater than one. In this case, the mantissa of the sum must be normalized; we will denote this rounding error by η_2.

Let the exact sum of $x_1 = 2^{b_1}(a_1)$ and $x_2 = 2^{b_2}(a_2)$ be $2^{b_3}(a_3)$, and assume that $x_1 \leq x_2$. Then

$$|\eta_1| \leq \frac{1}{2}\, 2^{-t}(2^{b_3-1})$$

and

$$|\eta_2| \leq \frac{1}{2}\, 2^{-t}(2^{b_3}) .$$

Therefore,

$$|\eta_1+\eta_2| \le |\eta_1| + |\eta_2|$$

$$\le \frac{3}{2}\, 2^{-t}(2^{b_3-1})$$

$$\le \frac{3}{2}\, 2^{-t}\, |x_1+x_2| \ ,$$

as the round-off error η_2 will not arise unless $|x_1+x_2| \ge \frac{1}{2}\, 2^{b_3}$. Since

$$fl_3(x_1+x_2) \equiv (x_1+x_2) + \eta_1+\eta_2 \ ,$$

if we write

$$fl_3(x_1+x_2) \equiv (x_1+x_2)(1+\varepsilon_1) \ ,$$

then

$$|\varepsilon_1| \le \frac{3}{2}\, 2^{-t} \ .$$

Q.E.D.

Lemma 2.2 *Using a double-precision accumulator but without floating-point accumulation,*

$$fl\Big(\sum_1^n a_i b_i\Big) \equiv \sum_1^n a_i b_i(1+\varepsilon_i) \ ,$$

where

$$(1-2^{-t})^n \le 1+\varepsilon_1 \le (1+2^{-t})^n$$

and

$$(1-2^{-t})^{n-r+2} \le 1+\varepsilon_r \le (1+2^{-t})^{n-r+2} ,$$

$$r=2,3,\ldots,n.$$

Proof: We define quantities s_r and p_r recursively by the relations

$$p_r = fl(a_r b_r)$$

and

$$s_1 = p_1, \quad s_r = fl(s_{r-1}+p_r) .$$

Then it follows from Theorem 2.1 that

$$p_r \equiv a_r b_r(1+\xi_r),$$

where

$$|\xi_r| \le 2^{-t} ,$$

and

$$s_r \equiv (s_{r-1}+p_r)(1+\eta_r) ,$$

where

$$|\eta_r| \leq 2^{-t} .$$

Hence

$$s_n \equiv \sum_{1}^{n} a_i b_i (1+\varepsilon_i) ,$$

where

$$1+\varepsilon_1 = (1+\xi_1)(1+\eta_2) \ldots (1+\eta_n)$$

and

$$1+\varepsilon_r = (1+\xi_r)(1+\eta_r) \ldots (1+\eta_n), \quad r=2,3,\ldots,n.$$

Thus

$$(1-2^{-t})^n \leq 1+\varepsilon_1 \leq (1+2^{-t})^n$$

and

$$(1-2^{-t})^{n-r+2} \leq 1+\varepsilon_r \leq (1+2^{-t})^{n-r+2}, \quad r=2,3,\ldots,n.$$

Q.E.D.

Theorem 2.2

$$(2.35)\qquad fl_2\Big(\sum_1^n a_i b_i\Big) - \Big(\sum_1^n a_i b_i\Big)(1+\varepsilon) \equiv \sum_1^n a_i b_i \varepsilon_i \ ,$$

where

$$(2.36)\qquad \begin{cases} |\varepsilon_1| \le \frac{3}{2}\, n\ 2^{-2t_2} \\ |\varepsilon_r| \le \frac{3}{2}\,(n-r+2)\ 2^{-2t_2} \\ 2^{-2t_2} = (1.06)\ 2^{-2t} \ . \end{cases}$$

Proof: It follows immediately from Lemmas 2.1 and 2.2 that

$$(2.37)\qquad fl_2\Big(\sum_1^n a_i b_i\Big) \equiv \Big[\sum_1^n a_i b_i (1+\varepsilon_i)\Big](1+\varepsilon) \ ,$$

where

$$(2.38)\qquad (1-\tfrac{3}{2}\,2^{-2t})^n \le 1+\varepsilon_1 \le (1+\tfrac{3}{2}\,2^{-2t})^n \ ,$$

and

$$(2.39)\qquad (1-\tfrac{3}{2}\,2^{-2t})^{n-r+2} \le 1+\varepsilon_r \le (1+\tfrac{3}{2}\,2^{-2t})^{n-r+2} \ ,$$

$$r=2,3,\ldots,n.$$

In the result of Lemma 2.2, the term 2^{-t} is replaced by 2^{-2t} as the products $a_i b_i$ are stored as double-precision numbers. The factor $\frac{3}{2}$ is introduced since n double-precision numbers $a_i b_i$ are added using double-precision arithmetic (replace "single-precision" by "double-precision" in Lemma 2.1). The factor $(1+\varepsilon)$ accounts for the rounding of the double-precision sum to single-precision.

If we restrict a number p such that

$$\frac{3}{2} p\, 2^{-2t} < 0.1 ,$$

then expansion of $(1 \pm \frac{3}{2} 2^{-2t})^p$ according to the binomial theorem leads to the results

$$(1+\tfrac{3}{2} 2^{-2t})^p < 1+\tfrac{3}{2} p(1.06)\, 2^{-2t}$$

and

$$(1-\tfrac{3}{2} 2^{-2t})^p > 1-\tfrac{3}{2} p(1.06)\, 2^{-2t} .$$

If we define t_2 by

$$(1.06)\, 2^{-2t} = 2^{-2t_2} ,$$

then (2.38) and (2.39) may be written as

(2.40) $$|\varepsilon_1| \leq \frac{3}{2}\, n2^{-2t_2}$$

and

(2.41) $$|\varepsilon_r| \leq \frac{3}{2}\,(n-r+2)\; 2^{-2t_2}\ ,$$

respectively, assuming that

$$\frac{3}{2}\, n2^{-2t} < 0.1\ .$$

(2.37) may be written

(2.42) $$fl_2\Big(\sum_1^n a_i b_i\Big) - \Big(\sum_1^n a_i b_i\Big)(1+\varepsilon) \equiv \Big(\sum_1^n a_i b_i \varepsilon_i\Big)(1+\varepsilon).$$

The factor 1.06 introduced in the definition of t_2 is sufficiently generous that the factor $(1+\varepsilon)$ on the right-hand side of (2.42) may be dropped. Thus

(2.43) $$fl_2\Big(\sum_1^n a_i b_i\Big) - \Big(\sum_1^n a_i b_i\Big)(1+\varepsilon) \equiv \sum_1^n a_i b_i \varepsilon_i\ .$$

Q.E.D.

Since the ε_i in Theorem 2.2 are of the order ε^2, they may be neglected, for a sufficiently large value of t. Then (2.43) becomes

$$(2.44) \qquad fl_2\left(\sum_1^n a_i b_i\right) \equiv \left(\sum_1^n a_i b_i\right)(1+\varepsilon) \ .$$

The neglecting of error terms of the order 2^{-2t} does not usually affect the over-all result significantly, since t is greater than 20 for most large computers today.

2.3.3 Matrix Operations The final bound we wish to mention is a bound on the round-off error incurred in a matrix multiplication. As each element of a matrix product is an inner product, equation (2.44) may be used to prove the following theorem.

Theorem 2.3 *If A and B are matrices conformable for multiplication, then*

$$(2.45) \qquad fl_2(AB) \equiv AB + E \ ,$$

where

$$(2.46) \qquad |E| \leq \varepsilon \ |A||B| \ .$$

Proof: Let $C = AB$. Then

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \ ,$$

where n is the common dimension of A and B, and by (2.44),

$$fl_2(c_{ij}) \equiv \left(\sum_{k=1}^{n} a_{ik}b_{kj}\right)(1+\varepsilon) \ .$$

Hence

$$fl_2(C) = fl_2(AB)$$

$$\equiv (1+\varepsilon)\ AB$$

$$\equiv AB + \varepsilon AB \ .$$

Therefore, a comparison of the above equation with (2.45) gives

$$|E| \leq \varepsilon\ |A||B| \ .$$

Q.E.D.

As a norm bound, (2.46) can be written

(2.47) $$\|E\| \leq \varepsilon\ \|A\|\|B\| \ .$$

Whenever a multiplication or division is performed using powers of 2 (in a binary machine), no round-off error is incurred. This is true as only the exponent need be changed, and not the mantissa. For example, in the computation $y = x\ \lambda\ 2^a$, where λ represents either of the floating-point operations of multiplication or division, the mantissa of y equals the mantissa of x, and the exponent of y is computed

from a and the exponent of x. Consequently, a matrix multiplication involving any permutation matrix does not produce any additional round-off error.

CHAPTER III

A NORM BOUND ON THE ERROR IN COMPUTING A REFLEXIVE GENERALIZED INVERSE

3.1 Computation of a Reflexive Generalized Inverse

An elimination method for computing a Reflexive Generalized Inverse G of an arbitrary m-by-n matrix is given by Rao [16]. We will assume that

$$m \leq n \ . \tag{3.1}$$

If $m > n$, we will consider the transpose of A, since it can easily be shown that a generalized inverse of A' is G'. A procedure for computing G is outlined below.

Partition A such that

$$A = [L|R] \ , \tag{3.2}$$

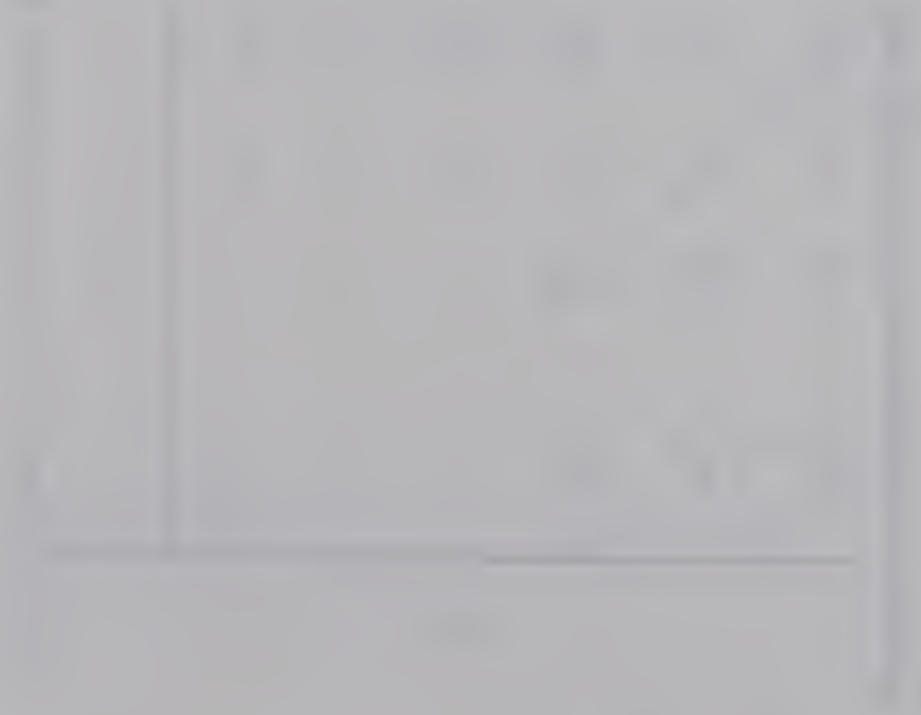
where L is a square matrix of order m and R is of dimension m-by-$(n-m)$. Then reduce L to an upper-triangular matrix by applying Gaussian forward elimination with complete pivoting to the rows of A, obtaining

$$(3.3)\quad A^{(1)} = \left[\begin{array}{cccccccccccc} d_1 & x & x & x & \cdots & x & x & \cdots & x & x & \cdots & x \\ 0 & d_2 & x & x & \cdots & x & x & \cdots & x & x & \cdots & x \\ 0 & 0 & d_3 & x & \cdots & x & x & \cdots & x & x & \cdots & x \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & d_r & x & \cdots & x & x & \cdots & x \\ \hline & & & & & & & & & & & \\ & & & & & & & & & & & \\ & & & & & & & 0 & & & & \end{array}\right],$$

where the elements x represent components which are not necessarily zero and where r, $r \leq m$, is the rank of A. Next apply Gaussian forward elimination to the columns of $A^{(1)}$ to obtain a matrix of the form

$$(3.4)\quad \Delta = \left[\begin{array}{cccccc|c} d_1 & 0 & 0 & 0 & \cdots & 0 & \\ 0 & d_2 & 0 & 0 & \cdots & 0 & \\ 0 & 0 & d_3 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \\ 0 & 0 & 0 & 0 & \cdots & d_r & \\ \hline & & & & & & \\ & & & 0 & & & \end{array}\right].$$

In practice, the forward elimination process which transforms $A^{(1)}$ into Δ need not be carried out, since we can simply write (3.4) from (3.3).



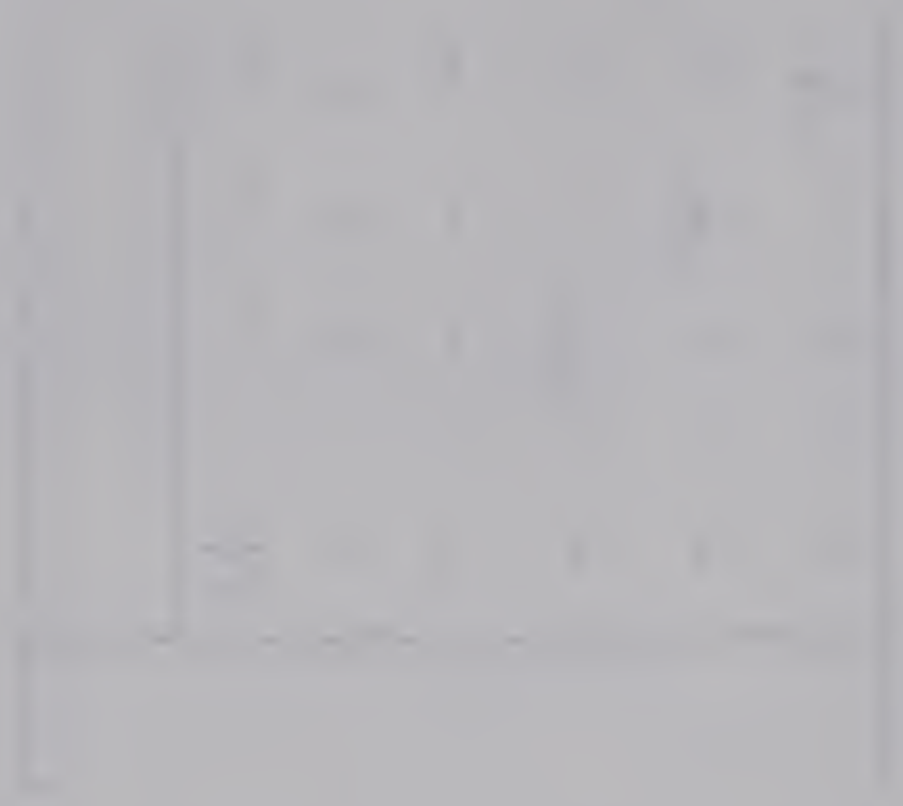
Let us define Δ^-, a matrix of dimension m-by-n, as

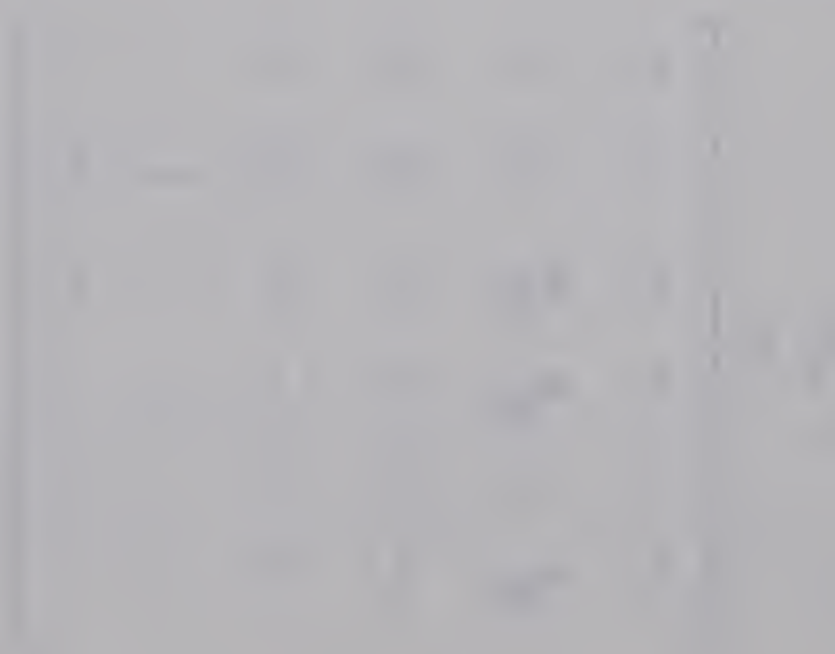
$$\Delta^- = \left[\begin{array}{cccccc|c} \frac{1}{d_1} & 0 & 0 & 0 & \dots & 0 & \\ 0 & \frac{1}{d_2} & 0 & 0 & \dots & 0 & \\ 0 & 0 & \frac{1}{d_3} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{d_r} & \\ \hline & & & 0 & & & \end{array}\right] . \tag{3.5}$$

Let us also define square matrices P_i, i=1,2,...,r, of order m as follows: each matrix P_i is identical to the unit matrix except for the elements in its i-th column below the main diagonal. These elements are the multipliers from the Gaussian forward elimination which transforms A into $A^{(1)}$. For example,

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & m_{32} & 1 & 0 & \dots & 0 \\ 0 & m_{42} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & m_{m2} & 0 & 0 & \dots & 1 \end{bmatrix} . \tag{3.6}$$

For $r = m$, $P_r = I$.





Similarly, let us define square matrices Q_i, i=1,2,...,r, of order n, such that each matrix Q_i is identical to the unit matrix except for the elements in the i-th row to the right of the main diagonal. These elements are the multipliers required for the Gaussian forward elimination in transforming $A^{(1)}$ into Δ. For example,

$$Q_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & m_{23} & m_{24} & \dots & m_{2n} \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix} . \tag{3.7}$$

For $m = n = r$, $Q_r = I$.

Let R_i and C_i, i=1,2,...,r, be permutation matrices of orders m and n, respectively, which accomplish the complete pivoting at the i-th stage of the forward elimination in reducing A to $A^{(1)}$. Let us define

$$P = P_r R_r P_{r-1} R_{r-1} \dots P_1 R_1 \tag{3.8}$$

and

$$Q = C_1 C_2 \dots C_r Q_1 Q_2 \dots Q_r . \tag{3.9}$$

Then P and Q are non-singular matrices such that

$$PAQ = \Delta. \tag{3.10}$$

Theorem 3.1 *A Reflexive Generalized Inverse of A is given by*

$$G = Q(\Delta^{-})'P \ . \tag{3.11}$$

Proof: By (3.10),

$$A = P^{-1}\Delta Q^{-1} \ . \tag{3.12}$$

Therefore,

$$\begin{aligned} AGA &= P^{-1}\Delta Q^{-1}Q(\Delta^{-})'PP^{-1}\Delta Q^{-1} \\ &= P^{-1}\Delta(\Delta^{-})'\Delta Q^{-1} \\ &= P^{-1}\Delta Q^{-1} \\ &= A. \end{aligned}$$

Also,

$$GAG = Q(\Delta^-)'PP^{-1}\Delta Q^{-1}Q(\Delta^-)'P$$

$$= Q(\Delta^-)'\Delta(\Delta^-)'P$$

$$= Q(\Delta^-)'P$$

$$= G.$$

Q.E.D.

3.2 An Error Analysis

Let F_1, F_2 and F_3 be the matrices of round-off errors incurred in the calculation of Q, $(\Delta^-)'$ and P, respectively. Then F_1 and F_3 are square and of orders n and m, respectively, and F_2 is of dimension n-by-m. The computational equation for the calculation of G becomes

$$(3.13) \qquad G \equiv \left[(Q+F_1)((\Delta^-)'+F_2)+F_4\right](P+F_3)+F_5$$

or

$$(3.14) \qquad G \equiv Q(\Delta^-)'P+\left[F_1((\Delta^-)'+F_2)+QF_2\right]P+(Q+F_1)((\Delta^-)'+F_2)F_3$$

$$+F_4(P+F_3)+F_5 \ ,$$

where F_4 and F_5 are the matrices of round-off errors due to the multiplications of Q by $(\Delta^-)'$ and $Q(\Delta^-)'$ by P, respectively. We will denote the exact value of G by G_E

and the computed value of G by G_C. Then it follows from (3.14) that

$$(3.15)\qquad |G_E-G_C| = \Big|\Big[F_1((\Delta^-)'+F_2)+QF_2\Big]P+(Q+F_1)((\Delta^-)'+F_2)F_3 + F_4(P+F_3)+F_5\Big| .$$

Thus we obtain the following norm bound on the error incurred in the calculation of G:

$$(3.16)\qquad \|G_E-G_C\| \le \Big[\|F_1\|(\|(\Delta^-)'\|+\|F_2\|)+\|Q\|\|F_2\|\Big]\|P\| + (\|Q\|+\|F_1\|)(\|(\Delta^-)'\|+\|F_2\|)\|F_3\| + \|F_4\|(\|P\|+\|F_3\|)+\|F_5\| .$$

We now proceed to place bounds on $\|F_i\|$, $i=1,2,3,4,5$. The following norm will be used, since it can easily be computed:

$$(3.17)\qquad \|A\|_\infty = \max_i \sum_j |a_{ij}| ,$$

where $A = (a_{ij})$ is an arbitrary square matrix. Equation (3.17) can be extended to include rectangular matrices, since these can be made square by bordering them with zeros (see Section 2.1).

3.2.1 A Bound on $\|F_1\|_\infty$ Let us denote the exact and computed values of Q by Q_E and Q_C, respectively. Therefore

$$|Q_E - Q_C| = |F_1| \quad . \tag{3.18}$$

Let

$$Q^{(1)} = Q_1 Q_2 \ldots Q_r \quad . \tag{3.19}$$

Then the calculation of Q_C from $Q^{(1)}$ is accomplished by appropriate row interchanges, and involves no arithmetic computations. Thus if we denote the exact and computed values of $Q^{(1)}$ by $Q_E^{(1)}$ and $Q_C^{(1)}$, respectively, it follows that

$$\|Q_E^{(1)} - Q_C^{(1)}\|_\infty = \|F_1\|_\infty \quad . \tag{3.20}$$

In order to place an upper bound on the round-off error incurred in the calculation of $Q_C^{(1)}$, let us define recursively matrices T_i as follows:

$$\begin{cases} T_1 = Q_r \\ T_i = Q_{r-i+1} T_{i-1}, \quad i=2,3,\ldots,r. \end{cases} \tag{3.21}$$

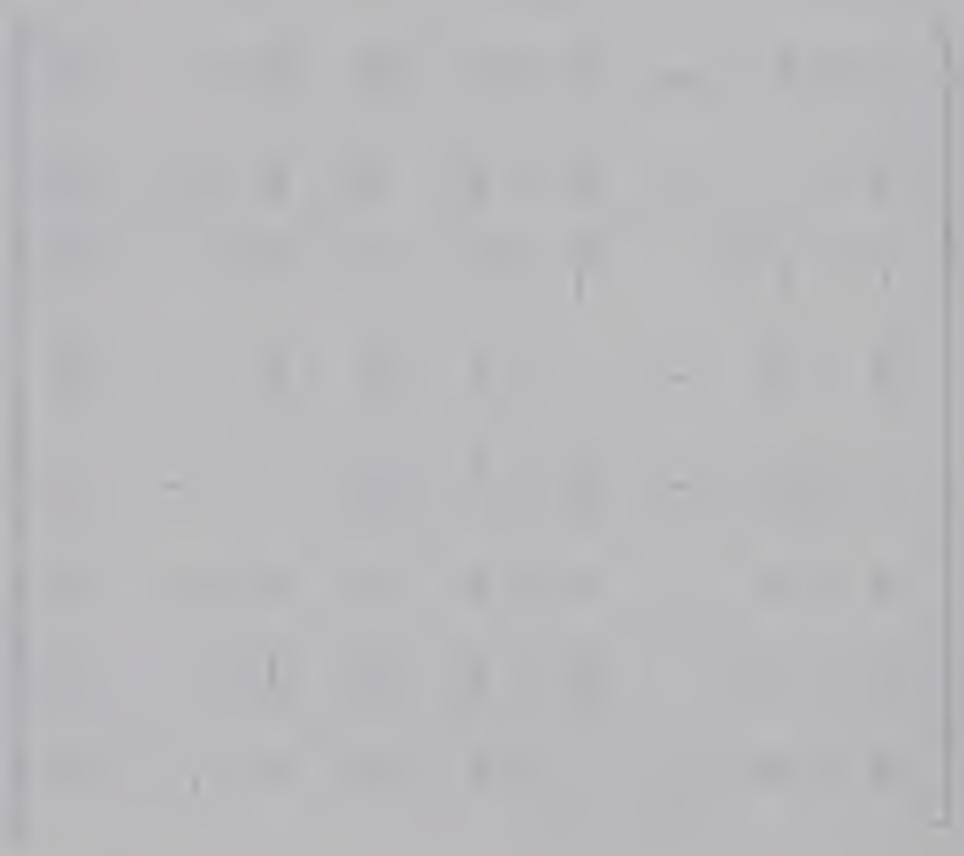
Then

$$T_r = Q^{(1)} . \tag{3.22}$$

First an upper bound on the error incurred in the matrix multiplication $Q_{r-i+1}T_{i-1}$ must be obtained. All of the elements of the matrices Q_i, $i=1,2,\ldots,r$, are less than or equal to one in magnitude since complete pivoting is used to transform A into $A^{(1)}$. Hence

$$|Q_i| \leq \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} , \tag{3.23}$$

where only the i-th row of the matrix in (3.23) contains more than one nonzero element. Using (3.23), it can easily be shown that the product $|Q_{r-i+1}||Q_{r-i+2}|\ldots|Q_r|$ is bounded by the matrix given in (3.24), and hence, $|T_i|$ is also so bounded.



$$
\left[\begin{array}{cccccccccc|ccc|ccc}
1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 1 & 1 & 2 & 4 & 8 & \cdots & 2^{i-3} & 2^{i-2} & 2^{i-1} & 2^{i-1} & \cdots & 2^{i-1} \\
0 & 0 & \cdots & 0 & 0 & 1 & 1 & 2 & 4 & \cdots & 2^{i-4} & 2^{i-3} & 2^{i-2} & 2^{i-2} & \cdots & 2^{i-2} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 1 & 1 & 2 & \cdots & 2^{i-5} & 2^{i-4} & 2^{i-3} & 2^{i-3} & \cdots & 2^{i-3} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 2 & 2 & \cdots & 2 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 & 1 & \cdots & 1 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 1
\end{array}\right] \tag{3.24}
$$

In the matrix (3.24), only rows $r-i+1$ to r, inclusively, contain more than one nonzero element.

Let

$$Q_{r-i+1} = (q_{ij}) ,$$

$$T_{i-1} = (t_{ij}) ,$$

and

$$Q_{r-i+1}T_{i-1} = T_i = (t_{ij}^{(1)}) .$$

From the structures of these matrices, it can be seen that

(3.25) T_i

$$
= \begin{bmatrix}
1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 1 & t^{(1)}_{r-i+1,r-i+2} & t^{(1)}_{r-i+1,r-i+3} & \cdots & t^{(1)}_{r-i+1,r} & t^{(1)}_{r-i+1,r+1} & \cdots & t^{(1)}_{r-i+1,n} \\
0 & 0 & \cdots & 0 & 0 & 1 & t_{r-i+2,r-i+3} & \cdots & t_{r-i+2,r} & t_{r-i+2,r+1} & \cdots & t_{r-i+2,n} \\
0 & 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & t_{r-i+3,r} & t_{r-i+3,r+1} & \cdots & t_{r-i+3,n} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & t_{r,r+1} & \cdots & t_{rn} \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 1
\end{bmatrix}.
$$

Thus, in the multiplication of T_{i-1} by Q_{r-i+1}, only the elements in row $r-i+1$ of T_i to the right of the main diagonal need be computed, the remainder being identical to the corresponding elements of T_{i-1}. Therefore, the matrix E_{i-1} of round-off errors for the multiplication of T_{i-1} by Q_{r-i+1} is null except for row $r-i+1$. Let

$$(3.26)\quad E_{i-1} = \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & e_{r-i+1,r-i+2} & e_{r-i+1,r-i+3} & \cdots & e_{r-i+1,n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

In (3.25), each element $t^{(1)}_{r-i+1,j}$, $j=r-i+2,r-i+3,\dots,n$, is computed as an inner product of a row vector of Q_{r-i+1} and a column vector of T_{i-1}. We now use the result of (2.44) to place a bound on the nonzero elements of (3.26). From (3.24) it can be seen that

$$(3.27)\qquad |t^{(1)}_{r-i+1,j}| \le 2^{i-1}, \quad j=r-i+2,r-i+3,\dots,n.$$

Therefore, we have

$$(3.28)\qquad |e_{r-i+1,j}| \le 2^{i-1}\varepsilon, \quad j=r-i+2,r-i+3,\dots,n.$$

The following theorem is a result of equation (3.28).

Theorem 3.2 *Let* Q_i, $i=1,2,\ldots,r$, *be square matrices of order* n *which transform* $A^{(1)}$ *into* Δ. *Define matrices* T_i, $i=1,2,\ldots,r$, *by (3.21). Then for*

$$T_i \equiv Q_{r+i+1}T_{i-1} + E_{i-1}\,, \tag{3.29}$$

we can bound the matrix E_{i-1} *of round-off errors as follows:*

$$\|E_{i-1}\|_\infty \le (n-r+i-1)2^{i-1}\varepsilon, \qquad i=2,3,\ldots,r. \tag{3.30}$$

$Q_C^{(1)}$ is calculated recursively as follows:

$$\begin{cases} T_1 = Q_r \\ T_2 \equiv Q_{r-1}Q_r + E_1 \\ T_3 \equiv Q_{r-2}Q_{r-1}Q_r + Q_{r-2}E_1 + E_2 \\ \vdots \\ T_r \equiv Q_1Q_2\cdots Q_r + Q_1Q_2\cdots Q_{r-2}E_1 + Q_1Q_2\cdots Q_{r-3}E_2 \\ \qquad + \ldots + Q_1Q_2E_{r-3} + Q_1E_{r-2} + E_{r-1} \end{cases} \tag{3.31}$$

and

$$T_r = Q_C^{(1)} \quad . \tag{3.32}$$

Therefore

$$|Q_E^{(1)} - Q_C^{(1)}| = |Q_1 Q_2 \cdots Q_{r-2} E_1 + Q_1 Q_2 \cdots Q_{r-3} E_2 + \cdots + Q_1 Q_2 E_{r-3} + Q_1 E_{r-2} + E_{r-1}| \quad . \tag{3.33}$$

Using (3.23), it may be shown that

(3.34) $|Q_1||Q_2|\ldots|Q_i|$

$$\leq \left[\begin{array}{ccccccccc|ccc}
1 & 1 & 2 & 4 & 8 & \ldots & 2^{i-3} & 2^{i-2} & 2^{i-1} & 2^{i-1} & \ldots & 2^{i-1} \\
0 & 1 & 1 & 2 & 4 & \ldots & 2^{i-4} & 2^{i-3} & 2^{i-2} & 2^{i-2} & \ldots & 2^{i-2} \\
0 & 0 & 1 & 1 & 2 & \ldots & 2^{i-5} & 2^{i-4} & 2^{i-3} & 2^{i-3} & \ldots & 2^{i-3} \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & 0 & 0 & 0 & \ldots & 1 & 1 & 2 & 2 & \ldots & 2 \\
0 & 0 & 0 & 0 & 0 & \ldots & 0 & 1 & 1 & 1 & \ldots & 1 \\
0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 1 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 1
\end{array}\right].$$

Hence,

$$\|Q_1\|_\infty\|Q_2\|_\infty\cdots\|Q_i\|_\infty \le n2^{i-1}\ , \quad i=1,2,\ldots,r\ . \tag{3.35}$$

Using (3.33),

$$\begin{aligned}\|Q_E^{(1)}-Q_C^{(1)}\|_\infty \le\ & \|Q_1\|_\infty\|Q_2\|_\infty\cdots\|Q_{r-2}\|_\infty\|E_1\|_\infty \\ & + \|Q_1\|_\infty\|Q_2\|_\infty\cdots\|Q_{r-3}\|_\infty\|E_2\|_\infty+\cdots+\|Q_1\|_\infty\|Q_2\|_\infty\|E_{r-3}\|_\infty \\ & + \|Q_1\|_\infty\|E_{r-2}\|_\infty+\|E_{r-1}\|_\infty\ .\end{aligned} \tag{3.36}$$

Using (3.30) and (3.35) in (3.36),

$$\begin{aligned}\|Q_E^{(1)}-Q_C^{(1)}\|_\infty &\le 2^{r-3}n(n-r+1)2\varepsilon \\ &\quad + 2^{r-4}n(n-r+2)4\varepsilon+\cdots+2n(n-3)2^{r-3}\varepsilon \\ &\quad + n(n-2)2^{r-2}\varepsilon + (n-1)2^{r-1}\varepsilon \\ &= n2^{r-2}\varepsilon\left[(n-r+1)+(n-r+2)\right. \\ &\quad \left.+\cdots+ (n-3)+(n-2)+\frac{2(n-1)}{n}\right] \\ &< 2^{r-3}n(r-1)(2n-r)\varepsilon\ ,\end{aligned} \tag{3.37}$$

where the last inequality holds only if $n \geq 2$. The following theorem is a result of equations (3.20) and (3.37).

Theorem 3.3 *Let* F_1 *be the matrix of round-off errors in the calculation of* Q *according to equation (3.9). Then*

$$\|F_1\|_\infty < 2^{r-3} n(r-1)(2n-r)\varepsilon \tag{3.38}$$

for $n \geq 2$.

3.2.2 A Smaller Bound on $\|F_1\|_\infty$ We now attempt to place a smaller norm bound on F_1 by bounding the right-hand side of equation (3.33), using (3.26) and (3.34). From (2.44) and (3.24), it follows that (3.26) may be bounded by

$$|E_{i-1}| \leq \varepsilon \left[\begin{array}{cccccccc|ccc} 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 1 & 2 & 4 & \cdots & 2^{i-2} & 2^{i-1} & 2^{i-1} & \cdots & 2^{i-1} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{array}\right], \tag{3.39}$$

where only the elements in the (r-i+1)-th row and columns r-i+2, r-i+3,...,n are nonzero. Direct multiplication of (3.34) and (3.39) produces the following result:

$$(3.40)\qquad |Q_1||Q_2| \ldots |Q_i||E_{r-i-1}|$$

$$\leq \varepsilon \left[\begin{array}{cccccccc|ccc}
0 & \ldots & 0 & 2^{i-1} & 2^{i} & 2^{i+1} & \ldots & 2^{r-2} & 2^{r-2} & \ldots & 2^{r-2} \\
0 & \ldots & 0 & 2^{i-2} & 2^{i-1} & 2^{i} & \ldots & 2^{r-3} & 2^{r-3} & \ldots & 2^{r-3} \\
\vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
0 & \ldots & 0 & 2 & 2^{2} & 2^{3} & \ldots & 2^{r-i} & 2^{r-i} & \ldots & 2^{r-i} \\
0 & \ldots & 0 & 1 & 2 & 2^{2} & \ldots & 2^{r-i-1} & 2^{r-i-1} & \ldots & 2^{r-i-1} \\
0 & \ldots & 0 & 1 & 2 & 2^{2} & \ldots & 2^{r-i-1} & 2^{r-i-1} & \ldots & 2^{r-i-1} \\
0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\
\vdots & & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
0 & \ldots & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & & 0
\end{array}\right],$$

where rows i+2,i+3,...,n and columns 1,2,...,i+1 are zero. Columns r+2,r+3,...,n are equal and are partitioned from the remainder of the matrix.

It follows from (3.33) that

$$(3.41)\qquad |Q_E^{(1)}-Q_C^{(1)}| \le |Q_1||Q_2| \ldots |Q_{r-2}||E_1|$$

$$+ |Q_1||Q_2| \ldots |Q_{r-3}||E_2|$$

$$+ \ldots + |Q_1||Q_2||E_{r-3}|$$

$$+ |Q_1||E_{r-2}| + |E_{r-1}| \quad .$$

By substituting (3.40) in (3.41) for $i=1,2,\ldots,r-2$, we obtain

(3.42) $|Q_E^{(1)} - Q_C^{(1)}|$

$$\leq \varepsilon \left[\begin{array}{ccccccc|ccc}
0 & 2(2^{-1}) & 3(2^{0}) & 4(2^{1}) & \cdots & (r-1)2^{r-4} & r2^{r-3} & r2^{r-2} & \cdots & r2^{r-2} \\
0 & 0 & 2(2^{-1}) & 3(2^{0}) & \cdots & (r-2)2^{r-5} & (r-1)2^{r-4} & (r-1)2^{r-3} & \cdots & (r-1)2^{r-3} \\
0 & 0 & 0 & 2(2^{-1}) & \cdots & (r-3)2^{r-6} & (r-2)2^{r-5} & (r-2)2^{r-4} & \cdots & (r-2)2^{r-4} \\
\vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\
0 & 0 & 0 & 0 & \cdots & 2(2^{-1}) & 3(2^{0}) & 3(2^{1}) & \cdots & 3(2^{1}) \\
0 & 0 & 0 & 0 & \cdots & 0 & 2(2^{-1}) & 2(2^{0}) & \cdots & 2(2^{0}) \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0
\end{array}\right].$$

Let the matrix on the right-hand side of (3.42) be denoted by $H = (h_{ij})$. Then

$$h_{ij} = \begin{cases} (j-i+1)\ 2^{j-i-2}\ \varepsilon, & i < j \le r \\ (r-i+1)\ 2^{r-i-1}\ \varepsilon, & j > r \text{ and } i \le r-1 \\ 0\ , \text{ otherwise.} \end{cases} \tag{3.43}$$

From (3.42), we have

$$\|H\|_\infty \le \varepsilon\left[r(2^{r-2}-\tfrac{1}{2}) + (n-r)r2^{r-2}\right] \tag{3.44}$$

$$< 2^{r-2}\ r(n-r+1)\ \varepsilon,$$

and therefore, can state

Theorem 3.4 *Let F_1 be the matrix of round-off errors in the calculation of Q in (3.9). Then*

$$\|F_1\|_\infty < 2^{r-2}\ r(n-r+1)\ \varepsilon. \tag{3.45}$$

This bound is at least as small as that given by (3.38) since

$$2(n-r+1) \le 2n-r$$

and

$$r \leq n(r-1)$$

for $r \geq 2$.

3.2.3 A Bound on $\|F_2\|_\infty$ If we denote the exact and computed values of $A^{(1)}$ by $A_E^{(1)}$ and $A_C^{(1)}$, respectively, then Wilkinson has shown in [21] that $A^{(1)}$ satisfies the computational equation

$$A_C^{(1)} \equiv A_E^{(1)} + E_1 \quad , \tag{3.46}$$

where

(3.47) $|E_1|$

$$\leq (2.01)g\varepsilon \left[\begin{array}{ccccccccc|ccc}
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\
1 & 1 & 1 & \ldots & 1 & 1 & 1 & \ldots & 1 & 1 & \ldots & 1 \\
1 & 2 & 2 & \ldots & 2 & 2 & 2 & \ldots & 2 & 2 & \ldots & 2 \\
1 & 2 & 3 & \ldots & 3 & 3 & 3 & \ldots & 3 & 3 & \ldots & 3 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
1 & 2 & 3 & \ldots & r-2 & r-2 & r-2 & \ldots & r-2 & r-2 & \ldots & r-2 \\
1 & 2 & 3 & \ldots & r-2 & r-1 & r-1 & \ldots & r-1 & r-1 & \ldots & r-1 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0
\end{array}\right],$$

where the left partition of the matrix in (3.47) is square and of order m and where g is the element of maximum modulus at any stage in the reduction of A to $A^{(1)}$.

The only round-off errors incurred in the computation of Δ are in the calculation of the elements d_i, $i=1,2,\ldots,r$, since all other elements of Δ may be set to zero. If Δ satisfies the computational equation

$$\Delta \equiv \Delta + E_2 \,, \tag{3.48}$$

then it follows from (3.47) that

$$|E_2| \leq (2.01)g\varepsilon \left[\begin{array}{ccccc|c} 0 & 0 & 0 & \ldots & 0 & \\ 0 & 1 & 0 & \ldots & 0 & \\ 0 & 0 & 2 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \\ 0 & 0 & 0 & \ldots & r-1 & \\ \hline & & & 0 & & \end{array}\right] \tag{3.49}$$

$$= \left[\begin{array}{c|c} \dfrac{E_3}{0} & 0 \end{array}\right] ,$$

where E_3 is a diagonal matrix of order r.

Let e_i, i=1,2,...,r, be the diagonal elements of E_3, i.e.

$$e_i = (i-1)(2.01)g\varepsilon \quad . \tag{3.50}$$

Therefore, if we denote the exact and computed values of the nonzero terms of Δ by $(d_i)_E$ and $(d_i)_C$, respectively, then

$$|(d_i)_E - (d_i)_C| \leq e_i \quad , \quad i=1,2,...,r. \tag{3.51}$$

From the computational equation

$$d_i \equiv d_i + e_i \quad , \tag{3.52}$$

we obtain

$$\frac{1}{d_i} \equiv \frac{1}{d_i+e_i} + e_i^{(1)} \quad , \tag{3.53}$$

where it is assumed that $d_i+e_i \neq 0$, and $e_i^{(1)}$ represents the round-off error due to a floating-point division. Hence, by Theorem 2.1,

$$e_i^{(1)} = \frac{\varepsilon}{d_i+e_i} \quad . \tag{3.54}$$

We may rewrite (3.53) as

$$\frac{1}{d_i} \equiv \frac{1}{d_i} - \frac{e_i}{d_i(d_i+e_i)} + e_i^{(1)} \quad . \tag{3.55}$$

Therefore, denoting the exact and computed values of $\frac{1}{d_i}$ by $(\frac{1}{d_i})_E$ and $(\frac{1}{d_i})_C$, respectively, we have that

$$\left|(\frac{1}{d_i})_E - (\frac{1}{d_i})_C\right| = \left|\frac{\varepsilon}{d_i+e_i} - \frac{e_i}{d_i(d_i+e_i)}\right|$$

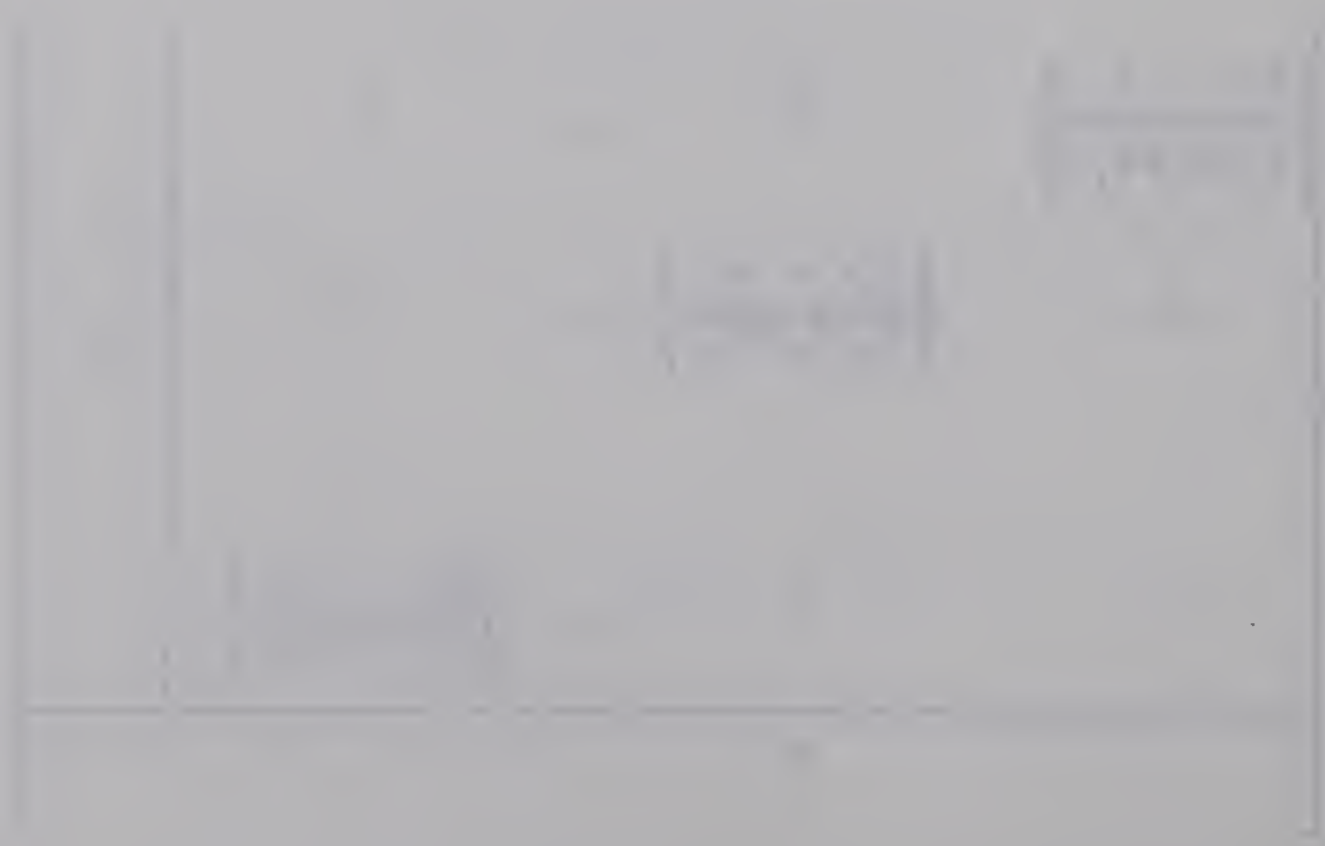
$$= \left|\frac{d_i\varepsilon - e_i}{d_i(d_i+e_i)}\right| \quad .$$

Thus, if

$$(\Delta^-)' \equiv (\Delta^-)' + F_2 \quad , \tag{3.56}$$

then

$$|F_2| \leq \begin{bmatrix} \left|\frac{d_1\varepsilon - e_1}{d_1(d_1+e_1)}\right| & 0 & \cdots & 0 & \\ 0 & \left|\frac{d_2\varepsilon - e_2}{d_2(d_2+e_2)}\right| & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \\ 0 & 0 & \cdots & \left|\frac{d_r\varepsilon - e_r}{d_r(d_r+e_r)}\right| & \\ \hline 0 & & & & \end{bmatrix} \quad . \tag{3.57}$$



The result of this section is stated as

Theorem 3.5 *Let* F_2 *be the matrix of round-off errors in the calculation of* $(\Delta^-)'$, *defined in (3.5). Then*

$$\|F_2\|_\infty \le \max_{1\le i\le r} \left[\left| \frac{d_i\varepsilon - e_i}{d_i(d_i+e_i)} \right| \right] . \tag{3.58}$$

3.2.4 A Bound on $\|F_3\|_\infty$ Let the exact and computed values of P, defined in (3.8), be P_E and P_C. Then

$$|P_E - P_C| = |F_3| . \tag{3.59}$$

Since the matrices R_i, i=1,2,...,r, are permutation matrices, there is no round-off error introduced due to a matrix multiplication involving any of the R_i (see Section 2.3).

Let the matrices S_i be defined recursively as follows:

$$\begin{cases} S_1 = P_r R_r \\ \\ S_i = S_{i-1} P_{r-i+1} R_{r-i+1}, \quad i=2,3,\ldots,r. \end{cases} \tag{3.60}$$

Then

$$S_r = P \ . \tag{3.61}$$

An upper bound for the round-off error incurred in the matrix multiplication $S_{i-1}P_{r-i+1}R_{r-i+1}$ is given by (2.47). If

$$S_{i-1}P_{r-i+1}R_{r-i+1} \equiv S_{i-1}P_{r-i+1}R_{r-i+1} + E_{r-i+1} \ , \tag{3.62}$$

then

$$\|E_{r-i+1}\|_\infty \le \|S_{i-1}\|_\infty \|P_{r-i+1}R_{r-i+1}\|_\infty \varepsilon, \quad i=2,3,\ldots,r. \tag{3.63}$$

From (3.60) and (3.62) we obtain the following set of computational equations for the calculation of P_C:

$$\begin{cases} S_1 = P_r R_r \\ S_2 \equiv P_r R_r P_{r-1} R_{r-1} + E_{r-1} \\ S_3 \equiv P_r R_r P_{r-1} R_{r-1} P_{r-2} R_{r-2} + E_{r-1} P_{r-2} R_{r-2} + E_{r-2} \\ \vdots \\ S_r \equiv P_r R_r P_{r-1} R_{r-1} \cdots P_1 R_1 + E_{r-1} P_{r-2} R_{r-2} P_{r-3} R_{r-3} \cdots P_1 R_1 \\ \qquad + E_{r-2} P_{r-3} R_{r-3} P_{r-4} R_{r-4} \cdots P_1 R_1 + \cdots + E_3 P_2 R_2 P_1 R_1 \\ \qquad + E_2 P_1 R_1 + E_1 \ . \end{cases} \tag{3.64}$$

Hence, from the final computational equation in (3.64), we obtain

$$\|F_3\|_\infty \le \|E_{r-1}P_{r-2}R_{r-2}P_{r-3}R_{r-3}\cdots P_1R_1\|_\infty + \|E_{r-2}P_{r-3}R_{r-3}P_{r-4}R_{r-4}\cdots P_1R_1\|_\infty + \ldots + \|E_3P_2R_2P_1R_1\|_\infty + \|E_2P_1R_1\|_\infty + \|E_1\|_\infty . \tag{3.65}$$

Using the property

$$\|AB\|_\infty \le \|A\|_\infty\|B\|_\infty \tag{3.66}$$

and the fact that

$$\|R_i\|_\infty = 1 , \qquad i=1,2,\ldots,r-1, \tag{3.67}$$

the inequality (3.63) becomes

$$\|E_{r-i+1}\|_\infty \le \|S_{i-1}\|_\infty\|P_{r-i+1}\|_\infty\varepsilon \le \|P_r\|_\infty\|P_{r-1}\|_\infty\cdots\|P_{r-i+2}\|_\infty\|P_{r-i+1}\|_\infty\varepsilon. \tag{3.68}$$

Using (3.66), (3.67) and (3.68), (3.65) becomes

$$\|F_3\|_\infty \leq (r-1)\|P_r\|_\infty\|P_{r-1}\|_\infty\cdots\|P_1\|_\infty\varepsilon \ . \tag{3.69}$$

Since all of the multipliers used in the construction of the matrices P_i, i=1,2,...,r, have the property that their moduli are less than or equal to one, we obtain

$$\|P_i\|_\infty \leq 2 \ . \tag{3.70}$$

The following theorem is a result of (3.69) and (3.70).

Theorem 3.6 *Let F_3 be the matrix of round-off errors in the calculation of P according to equation (3.8). Then*

$$\|F_3\|_\infty \leq (r-1)2^r\varepsilon \ . \tag{3.71}$$

3.2.5 A Bound for the Computed Reflexive Generalized Inverse A norm bound for the round-off error incurred in computing a Reflexive Generalized Inverse can be obtained from (3.16). It follows from (2.47) that

$$\|F_4\|_\infty \leq \varepsilon\|Q\|_\infty\|(\Delta^-)'\|_\infty \tag{3.72}$$

and

$$\|F_5\|_\infty \leq \varepsilon\|Q\|_\infty\|(\Delta^-)'\|_\infty\|P\|_\infty \quad . \tag{3.73}$$

Substitution of (3.45), (3.71), (3.72) and (3.73) into (3.16) gives the following norm bound:

$$\begin{aligned}\|G_E - G_C\|_\infty &\leq \Big[r(n-r+1)2^{r-2}(\|(\Delta^-)'\|_\infty + \|F_2\|_\infty)\varepsilon \\ &\quad + \|Q\|_\infty\|F_2\|_\infty\Big]\|P\|_\infty + (r-1)2^r\varepsilon \\ &\quad \times(\|(\Delta^-)'\|_\infty + \|F_2\|_\infty)\Big[\|Q\|_\infty + r(n-r+1)2^{r-2}\varepsilon\Big] \\ &\quad + 2\varepsilon\|Q\|_\infty\|(\Delta^-)'\|_\infty\|P\|_\infty \ ,\end{aligned} \tag{3.74}$$

where $\|F_2\|_\infty$ is bounded in (3.58). The product $\|F_4\|_\infty\|F_3\|_\infty$ is neglected as it is of the order ε^2. Our computer program for the computation of the Reflexive Generalized Inverse includes the evaluation of the right-hand side of (3.74) (see Chapter V).

CHAPTER IV

A NORM BOUND ON THE ERROR IN COMPUTING THE PSEUDOINVERSE

4.1 Computation of the Pseudoinverse

We will compute the Pseudoinverse A^+ of an arbitrary complex matrix A of dimension m-by-n according to the algorithm given by Willner [23]. Throughout this chapter, it is assumed that the rank of A is r and that A_j, $j=1,2,\ldots,n$, is the j-th column of A. Also, let e_i, $i=1,2,\ldots,m$, be the i-th unit vector of the identity matrix of order m. The proof of Willner's algorithm requires the following lemma, which is due to Greville [7].

Lemma 4.1 *If a matrix A of dimension m-by-n and rank r can be expressed as a product*

$$A = PB \ ,$$

where P and B are matrices of dimension m-by-r and r-by-n, respectively, and both are of rank r, then

$$A^+ = B^*(BB^*)^{-1}(P^*P)^{-1}P^* \ . \tag{4.1}$$

Proof: To prove the above, it suffices to show that equations (1.1), (1.2), (1.3) and (1.4) are satisfied.

$$AA^{+}A = PBB^{*}(BB^{*})^{-1}(P^{*}P)^{-1}P^{*}PB$$

$$= PB$$

$$= A \ .$$

$$A^{+}AA^{+} = B^{*}(BB^{*})^{-1}(P^{*}P)^{-1}P^{*}PBB^{*}(BB^{*})^{-1}(P^{*}P)^{-1}P^{*}$$

$$= B^{*}(BB^{*})^{-1}(P^{*}P)^{-1}P^{*}$$

$$= A^{+} \ .$$

$$(A^{+}A)^{*} = [B^{*}(BB^{*})^{-1}(P^{*}P)^{-1}P^{*}PB]^{*}$$

$$= [B^{*}(BB^{*})^{-1}B]^{*}$$

$$= B^{*}[(BB^{*})^{-1}]^{*}B$$

$$= B^{*}[(BB^{*})^{*}]^{-1}B$$

$$= B^{*}(BB^{*})^{-1}B$$

$$= B^{*}(BB^{*})^{-1}(P^{*}P)^{-1}P^{*}PB$$

$$= A^{+}A \ .$$

$$(AA^{+})^{*} = [PBB^{*}(BB^{*})^{-1}(P^{*}P)^{-1}P^{*}]^{*}$$

$$= [P(P^{*}P)^{-1}P^{*}]^{*}$$

$$= P[(P^{*}P)^{-1}]^{*}P^{*}$$

$$= P[(P^{*}P)^{*}]^{-1}P^{*}$$

$$= P(P^{*}P)^{-1}P^{*}$$

$$= PBB^{*}(BB^{*})^{-1}(P^{*}P)^{-1}P^{*}$$

$$= AA^{+} \ .$$

Q.E.D.

Theorem 4.1 *Let* A *be an arbitrary complex matrix of dimension* m-*by*-n *and rank* r. *Let* $H = \begin{bmatrix} B \\ O \end{bmatrix}$ *be the Hermite normal form of* A, *and let* Q^{-1} *be the nonsingular matrix of order* m *satisfying*

$$H = Q^{-1}A \ . \tag{4.2}$$

If Q *is partitioned as* $Q = [P|R]$, *where* P *is of dimension* m-*by*-r, *then*

$$A^{+} = B^{*}(P^{*}AB^{*})^{-1}P^{*} \ . \tag{4.3}$$

Proof: Since A is of rank r, it follows that B is of dimension r-by-n. The existence of the nonsingular matrix Q^{-1}, satisfying (4.2), follows from the fact that A can be reduced to H by elementary row operations. From (4.2),

$$\begin{aligned} A &= QH \\ &= [P|R] \begin{bmatrix} B \\ O \end{bmatrix} \\ &= PB \ . \end{aligned}$$

As A, P and B are all of rank r, we can use (4.1) to compute A^{+}. Therefore,

$$A^+ = B^*(BB^*)^{-1}(P^*P)^{-1}P^*$$

$$= B^*(P^*PBB^*)^{-1}P^*$$

$$= B^*(P^*AB^*)^{-1}P^* \ .$$

Q.E.D.

Willner's Algorithm Willner summarizes the computation of A^+ using (4.3) in the following five steps:

(i) Compute H by using Gauss-Jordan transformations.

(ii) From H determine P as follows: the i-th column of P, P_i, i=1,2,...,r, is equal to A_j if $H_j = e_i$, j=1,2,...,n.

(iii) Calculate P^*AB^*.

(iv) Invert P^*AB^*.

(v) Calculate A^+ using (4.3).

We now proceed to place an upper bound on the round-off error incurred in computing A^+ using this algorithm.

4.2 An Error Analysis

4.2.1 A Bound for the Computed Hermite Normal Form

The algorithm due to Marcus and Minc (see Section 2.1) for the computation of the Hermite normal form of A employs Type II elementary row operations. As the elements both above and below the pivotal element are reduced to zero,

these elementary row operations are equivalent to a Gauss-Jordan transformation, i.e., pre-multiplication of A, or a reduced form of A, by a matrix J_i as defined by (2.23). As A is of rank r, r Jordan transformations are required to reduce A to its Hermite normal form. As in Section 2.1, we will let n_i, $i=1,2,\ldots,r$, denote the column numbers of the r columns of A which contain pivotal elements for the Jordan transformations. These are not necessarily the first r columns of A. If we denote the matrix A after the i-th Jordan transformation by $A^{(i)} = (a_{jk}^{(i)})$, then the multipliers used to construct J_i will be defined by

$$m_{ji} = -\frac{a_{j,n_i}^{(i-1)}}{a_{i,n_i}^{(i-1)}}, \qquad \begin{matrix} i=1,2,\ldots,r, \\ j=1,2,\ldots,m, \quad j\neq i, \end{matrix} \tag{4.4}$$

rather than by (2.24). It is assumed that $A^{(0)} = A$.

As all of the pivotal elements of the Hermite normal form are equal to one, the i-th row of $A^{(i)}$ must be divided by the pivotal element $a_{i,n_i}^{(i)}$. This may be accomplished by pre-multiplication of $A^{(i)}$ by a diagonal matrix $D_i = (d_{jk})$, where

$$(4.5) \qquad D_i = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & & \frac{1}{a_{i,n_i}^{(i)}} & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{bmatrix}$$

and $d_{ii} = \frac{1}{a_{i,n_i}^{(i)}}$.

If we let

$$(4.6) \qquad J_i^{(1)} = D_i J_i \quad ,$$

then

$$(4.7)\qquad J_i^{(1)} = \begin{bmatrix} 1 & 0 & \dots & 0 & m_{1i} & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & m_{2i} & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & m_{i-1,i} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{a_{i,n_i}^{(i)}} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & m_{i+1,i} & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & m_{mi} & 0 & \dots & 1 \end{bmatrix} .$$

Therefore, if H denotes the Hermite Normal form of A, then

$$(4.8)\qquad \begin{aligned} H &= D_r J_r D_{r-1} J_{r-1} \dots D_1 J_1 A \\ &= J_r^{(1)} J_{r-1}^{(1)} \dots J_1^{(1)} A \ . \end{aligned}$$

The intermediate matrices are given by

$$(4.9)\qquad \begin{aligned} A^{(i)} &= J_i^{(1)} J_{i-1}^{(1)} \dots J_1^{(1)} A \\ &= J_i^{(1)} A^{(i-1)} , \qquad i=1,2,\dots,r, \end{aligned}$$

and

$$A^{(r)} = H \ . \tag{4.10}$$

Complete pivoting may not be used since H must be obtained from A by elementary row operations only. Partial pivoting, however, may be used to insure that those elements of $J_i^{(1)}$ below the main diagonal are less than or equal to one in magnitude.

It is possible to multiply the pivotal row of $A^{(i)}$, $i=0,1,\ldots,r-1$, by a power of 2 (in a binary machine) to insure that all elements of $J_{i+1}^{(1)}$ will be less than or equal to one in magnitude. We let P_i, $i=0,1,\ldots,r-1$, be equal to a unit matrix except for its $(i+1)$-th diagonal element, which is some non-negative power of 2, i.e.

$$P_i = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ & & & & p_{i+1,i+1}^{(i)} & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{bmatrix} \tag{4.11}$$

where

$$p^{(i)}_{i+1,i+1} = 2^{q_i} . \tag{4.12}$$

q_i is chosen so that the modulus of the product of 2^{q_i} and the pivotal element of $A^{(i)}$, $a^{(i)}_{i+1,n_{i+1}}$, is greater than or equal to $|a^{(i)}_{j,n_{i+1}}|$, $j=1,2,\ldots,m$, $j\neq i+1$. Then

$$H = J^{(1)}_r P_{r-1} J^{(1)}_{r-1} P_{r-2} \ldots J^{(1)}_1 P_0 A , \tag{4.13}$$

where all elements of each matrix $J^{(1)}_i$, $i=1,2,\ldots,r$, are less than or equal to one in magnitude.

Since $\|P_i\|_\infty$ is unbounded, this does not, however, lower the upper bound for the round-off error in the calculation of H. Hence, we will use (4.8) for the computation of H.

If we let E_i, $i=1,2,\ldots,r$, be the matrix of round-off errors incurred in multiplying $A^{(i-1)}$ by $J^{(1)}_i$, then the computed value of H, H_C, is defined by the following set of computational equations:

(4.14)
$$
\begin{cases}
A^{(1)} \equiv J_1^{(1)} A + E_1 \\
A^{(2)} \equiv J_2^{(1)} J_1^{(1)} A + J_2^{(1)} E_1 + E_2 \\
\vdots \\
A^{(r)} \equiv J_r^{(1)} J_{r-1}^{(1)} \ldots J_1^{(1)} A \\
\qquad + J_r^{(1)} J_{r-1}^{(1)} \ldots J_2^{(1)} E_1 \\
\qquad + J_r^{(1)} J_{r-1}^{(1)} \ldots J_3^{(1)} E_2 + \ldots \\
\qquad + J_r^{(1)} E_{r-1} + E_r \, .
\end{cases}
$$

Since (4.8) defines the exact value, H_E, of H, it follows from the last equation in (4.14) that

(4.15)
$$
\begin{aligned}
\| H_E - H_C \|_\infty = \| & J_r^{(1)} J_{r-1}^{(1)} \ldots J_2^{(1)} E_1 \\
& + J_r^{(1)} J_{r-1}^{(1)} \ldots J_3^{(1)} E_2 + \ldots \\
& + J_r^{(1)} E_{r-1} + E_r \|_\infty \\
\leq & \| J_r^{(1)} \|_\infty \| J_{r-1}^{(1)} \|_\infty \ldots \| J_2^{(1)} \|_\infty \| E_1 \|_\infty \\
& + \| J_r^{(1)} \|_\infty \| J_{r-1}^{(1)} \|_\infty \ldots \| J_3^{(1)} \|_\infty \| E_2 \|_\infty \\
& + \ldots + \| J_r^{(1)} \|_\infty \| E_{r-1} \|_\infty + \| E_r \|_\infty \, .
\end{aligned}
$$

By (2.47),

$$\|E_i\|_\infty \le \varepsilon \|J_i^{(1)}\|_\infty \|A^{(i-1)}\|_\infty , \qquad i=1,2,\ldots,r . \tag{4.16}$$

Therefore, using (4.9),

$$\|E_i\|_\infty \le \varepsilon \|J_i^{(1)}\|_\infty \|J_{i-1}^{(1)}\|_\infty \ldots \|J_1^{(1)}\|_\infty \|A\|_\infty , \quad i=1,2,\ldots,r. \tag{4.17}$$

Substituting (4.17) in (4.15), we obtain

$$\|H_E - H_C\|_\infty \le r\varepsilon \|J_r^{(1)}\|_\infty \|J_{r-1}^{(1)}\|_\infty \ldots \|J_1^{(1)}\|_\infty \|A\|_\infty . \tag{4.18}$$

Thus we may state

Theorem 4.2 *A norm bound on the round-off error incurred in computing the Hermite normal form of* A *is*

$$\|H_E - H_C\|_\infty \le r\varepsilon \|J_r^{(1)}\|_\infty \|J_{r-1}^{(1)}\|_\infty \ldots \|J_1^{(1)}\|_\infty \|A\|_\infty ,$$

where r *is the rank of* A *and the matrices* $J_i^{(1)}$ *are defined by (4.7).*

4.2.2 A Bound for the Product P*AB* The determination of the matrix P by Willner's algorithm involves no arithmetic calculations. Therefore no round-off errors are introduced,

and the elements of P are exact (to the precision of the computer).

As the initial step in bounding the round-off error incurred in computing the matrix product P^*AB^*, we consider the computation of P^*A. If F_1 is the matrix of round-off errors for the multiplication of P^* by A, then

$$P^*A \equiv P^*A + F_1 , \tag{4.19}$$

and by (2.47),

$$\|F_1\|_\infty \leq \varepsilon \|P^*\|_\infty \|A\|_\infty . \tag{4.20}$$

The elements of B^* are not exact, as are the elements of A and P^*, because B is a submatrix of H. It follows from (4.14) that

$$\begin{aligned} |H_E^* - H_C^*| = |&(J_r^{(1)} J_{r-1}^{(1)} \cdots J_2^{(1)} E_1 \\ &+ J_r^{(1)} J_{r-1}^{(1)} \cdots J_3^{(1)} E_2 + \cdots \\ &+ J_r^{(1)} E_{r-1} + E_r)^*| \end{aligned}$$

or

$$(4.21)\qquad |H_E^*-H_C^*| = |E_1^*J_2^{(1)*}J_3^{(1)*}\ \ldots\ J_r^{(1)*} + E_2^*J_3^{(1)*}J_4^{(1)*}\ \ldots\ J_r^{(1)*} + \ldots + E_{r-1}^*J_r^{(1)*} + E_r^*| \ .$$

Therefore,

$$(4.22)\qquad |H_C^*| \le |H_E^*| + F_2 \ ,$$

where

$$(4.23)\qquad F_2 = |E_1^*J_2^{(1)*}J_3^{(1)*}\ \ldots\ J_r^{(1)*} + E_2^*J_3^{(1)*}J_4^{(1)*}\ \ldots\ J_r^{(1)*} + \ldots + E_{r-1}^*J_r^{(1)*} + E_r^*| \ .$$

Thus

$$(4.24)\qquad \|H_C^*\|_\infty \le \|H_E^*\|_\infty + \|F_2\|_\infty \ .$$

We will denote the exact and computed values of B by B_E and B_C, respectively. Since B_C^* is a submatrix of H_C^*, and $\|H_E^*\|_\infty = \|B_E^*\|_\infty$, it follows that

$$\|B_C^*\|_\infty \le \|B_E^*\|_\infty + \|F_2\|_\infty \ . \tag{4.25}$$

The corresponding computational equation is

$$B^* \equiv B^* + F_2 \ , \tag{4.26}$$

where

$$\begin{aligned}
\|F_2\|_\infty \le\ & \|E_1^*\|_\infty \|J_2^{(1)*}\|_\infty \|J_3^{(1)*}\|_\infty \ \dots\ \|J_r^{(1)*}\|_\infty \\
& + \|E_2^*\|_\infty \|J_3^{(1)*}\|_\infty \|J_4^{(1)*}\|_\infty \ \dots\ \|J_r^{(1)*}\|_\infty \\
& + \ \dots\ + \|E_{r-1}^*\|_\infty \|J_r^{(1)*}\|_\infty + \|E_r^*\|_\infty \\
=\ & \|E_1\|_1 \|J_2^{(1)}\|_1 \|J_3^{(1)}\|_1 \ \dots\ \|J_r^{(1)}\|_1 \\
& + \|E_2\|_1 \|J_3^{(1)}\|_1 \|J_4^{(1)}\|_1 \ \dots\ \|J_r^{(1)}\|_1 \\
& + \ \dots\ + \|E_{r-1}\|_1 \|J_r^{(1)}\|_1 + \|E_r\|_1 \\
\le\ & r\varepsilon \|J_r^{(1)}\|_1 \|J_{r-1}^{(1)}\|_1 \ \dots\ \|J_1^{(1)}\|_1 \ .
\end{aligned} \tag{4.27}$$

The computational equation for the product P^*AB^*, using (4.19) and (4.26), can now be written as

$$(4.28) \qquad P^*AB^* \equiv (P^*A+F_1)(B^*+F_2) + F_3$$

or

$$(4.29) \qquad P^*AB^* \equiv P^*AB^* + P^*AF_2 + F_1B^* + F_1F_2 + F_3 ,$$

where F_3 is the matrix of round-off errors incurred in the multiplication of P^*A by B^*. The term F_1F_2 and the round-off error incurred in forming the products P^*AF_2 and F_1B^* will be neglected since they are of the order ε^2. By (2.47),

$$(4.30) \qquad \|F_3\|_\infty \leq \varepsilon \|P^*A\|_\infty \|B^*\|_\infty .$$

The following theorem is a result of (4.29) and the above discussion.

Theorem 4.3 *If*

$$(4.31) \qquad P^*AB^* \equiv P^*AB^* + F ,$$

then

$$(4.32) \qquad \|F\|_\infty \leq \|P^*\|_\infty \|A\|_\infty \|F_2\|_\infty + \|F_1\|_\infty \|B^*\|_\infty + \|F_3\|_\infty ,$$

where the bounds on $\|F_1\|_\infty$, $\|F_2\|_\infty$ *and* $\|F_3\|_\infty$ *are given by (4.20), (4.27) and (4.30), respectively.*

4.2.3 A Bound in Computing $(P^*AB^*)^{-1}$ The fourth step in Willner's algorithm for the computation of A^+ is the inversion of the square matrix P^*AB^* of order r. For the computed inverse, X_C^{-1}, of a nonsingular matrix X of order n, Wilkinson [21] has shown that

$$XX_C^{-1} - I = K \ , \tag{4.33}$$

where

$$\|K\|_\infty \leq \|X_C^{-1}\|_\infty \, g(2.005\, n^2+n^3)\varepsilon \ , \tag{4.34}$$

with terms of order ε^2 ignored. g denotes the element of maximum modulus at any stage in the reduction of X to X_C^{-1}. Therefore,

$$(P^*AB^*+F)(P^*AB^*+F)_C^{-1} - I = K \ , \tag{4.35}$$

where

$$\|K\|_\infty \leq \|(P^*AB^*+F)_C^{-1}\|_\infty \, g(2.005r^2+r^3)\varepsilon \ . \tag{4.36}$$

Hence

$$(4.37) \qquad (P^*AB^*+F)_C^{-1} = (P^*AB^*+F)_E^{-1}(I+K) \ ,$$

where $(P^*AB^*+F)_E^{-1}$ denotes the exact inverse of (P^*AB^*+F).

Wang [19] has shown that the inverse of a modified matrix $Y+E$ can be obtained from

$$(4.38) \qquad (Y+E)^{-1} = (I+Y^{-1}E)^{-1}Y^{-1} \ .$$

Using (4.38) in (4.37), we obtain

$$\begin{aligned}(4.39) \quad (P^*AB^*+F)_C^{-1} &= [I+(P^*AB^*)_E^{-1}F]^{-1}(P^*AB^*)_E^{-1}(I+K) \\ &= (P^*AB^*)_E^{-1} + [I+(P^*AB^*)_E^{-1}F]^{-1}(P^*AB^*)_E^{-1}(I+K) \\ &\quad - (P^*AB^*)_E^{-1} \\ &= (P^*AB^*)_E^{-1} + [I+(P^*AB^*)_E^{-1}F]^{-1}(P^*AB^*)_E^{-1} \\ &\quad \times[K-F(P^*AB^*)_E^{-1}] \\ &= (P^*AB^*)_E^{-1} + (P^*AB^*+F)_E^{-1}[K-F(P^*AB^*)_E^{-1}].\end{aligned}$$

From (4.39), we can write the computational equation

$$(4.40) \qquad (P^*AB^*)_C^{-1} \equiv (P^*AB^*)_E^{-1} + F_4 \ ,$$

where

$$(4.41) \qquad \|F_4\|_\infty \leq \|(P^*AB^*+F)_E^{-1}\|_\infty \|K-F(P^*AB^*)_E^{-1}\|_\infty \ .$$

From (4.39), it also follows that

$$(4.42) \qquad \|(P^*AB^*+F)_C^{-1}\|_\infty \leq \|(P^*AB^*)_E^{-1}\|_\infty + \|(P^*AB^*+F)_E^{-1}\|_\infty \|K-F(P^*AB^*)_E^{-1}\|_\infty .$$

Substituting (4.36) in (4.42), we obtain

$$(4.43) \qquad \|(P^*AB^*+F)_C^{-1}\|_\infty \leq \|(P^*AB^*)_E^{-1}\|_\infty + \|(P^*AB^*+F)_E^{-1}\|_\infty \Big[c\|(P^*AB^*+F)_C^{-1}\|_\infty + \|F\|_\infty \|(P^*AB^*)_E^{-1}\|_\infty\Big],$$

where

$$(4.44) \qquad c = g(2.005r^2+r^3)\varepsilon \ .$$

Assuming that the elements of $|F|$ are small relative to those of $|P^*AB^*|$, and that the matrix P^*AB^* is well-conditioned, the exact and computed values of $\|(P^*AB^*+F)^{-1}\|_\infty$ are approximately equal. In this case, (4.43) becomes

$$(4.45)\qquad -c\|(P^*AB^*+F)^{-1}\|_\infty^2 + \left[1-\|F\|_\infty\|(P^*AB^*)_E^{-1}\|_\infty\right] \times \|(P^*AB^*+F)^{-1}\|_\infty - \|(P^*AB^*)_E^{-1}\|_\infty \le 0 .$$

The discriminant of this quadratic expression in the variable $\|(P^*AB^*+F)^{-1}\|_\infty$ is

$$(4.46)\qquad d = (1-\|F\|_\infty\|(P^*AB^*)_E^{-1}\|_\infty)^2 - 4c\|(P^*AB^*)_E^{-1}\|_\infty,$$

which we can assume to be positive since c is of the order ε. It follows that the roots are either both positive or both negative, since the value of the quadratic expression is negative for $\|(P^*AB^*+F)^{-1}\|_\infty = 0$. To place an upper bound on $\|(P^*AB^*+F)^{-1}\|_\infty$, using (4.45), we need consider only the case of two positive roots. For

$$(4.47)\qquad 1 - \|F\|_\infty\|(P^*AB^*)_E^{-1}\|_\infty > 0 ,$$

both roots will be positive, and the smaller of the two roots

will equal

$$\frac{-1 + \|F\|_\infty \|(P^*AB^*)_E^{-1}\|_\infty + d^{1/2}}{-2c} \quad . \tag{4.48}$$

Then (4.45) implies that

$$\|(P^*AB^*+F)^{-1}\|_\infty \leq \frac{1 - \|F\|_\infty \|(P^*AB^*)_E^{-1}\|_\infty - d^{1/2}}{2c} \quad , \tag{4.49}$$

and from (4.36) and (4.41), we have that

$$\|F_4\|_\infty \leq \|(P^*AB^*+F)^{-1}\|_\infty \left[c\|(P^*AB^*+F)^{-1}\|_\infty + \|F\|_\infty \|(P^*AB^*)_E^{-1}\|_\infty \right] . \tag{4.50}$$

The result of this section is summarized in the following theorem.

Theorem 4.4 *Given that*

$$P^*AB^* \equiv P^*AB^* + F \ ,$$

where $\|F\|_\infty$ *is bounded by (4.32), then*

$$(P^*AB^*)^{-1} \equiv (P^*AB^*)^{-1} + F_4 \ ,$$

where $\|F_4\|_\infty$ *is bounded by (4.50).*

4.2.4 A Bound for the Computed Pseudoinverse The final step in Willner's algorithm for the computation of A^+ is the calculation of the matrix product $B^*(P^*AB^*)^{-1}P^*$. Considering first the multiplication of B^* by $(P^*AB^*)^{-1}$, it follows from (4.26) and (4.40) that

$$B^*(P^*AB^*)^{-1} \equiv (B^*+F_2)[(P^*AB^*)^{-1} + F_4] + F_5 \tag{4.51}$$

or

$$B^*(P^*AB^*)^{-1} \equiv B^*(P^*AB^*)^{-1} + B^*F_4 + F_2(P^*AB^*)^{-1} + F_2F_4 + F_5 \ , \tag{4.52}$$

where F_5 is the matrix of round-off errors in the multiplication of B^* by $(P^*AB^*)^{-1}$. Since the product F_2F_4 and the round-off errors incurred in forming the products B^*F_4 and $F_2(P^*AB^*)^{-1}$ are of order ε^2, they are neglected. Therefore,

$$B^*(P^*AB^*)^{-1} \equiv B^*(P^*AB^*)^{-1} + B^*F_4 + F_2(P^*AB^*)^{-1} + F_5 \ , \tag{4.53}$$

where, by (2.47),

$$\|F_5\|_\infty \le \varepsilon\|B^*\|_\infty\|(P^*AB^*)^{-1}\|_\infty \ . \tag{4.54}$$

Post-multiplication of (4.53) by P^* produces

$$\begin{aligned} B^*(P^*AB^*)^{-1}P^* \equiv\ & B^*(P^*AB^*)^{-1}P^* \\ & + B^*F_4P^* + F_2(P^*AB^*)^{-1}P^* \\ & + F_5P^* + F_6 \ , \end{aligned} \tag{4.55}$$

where F_6 is the matrix of round-off errors incurred in the multiplication of $B^*(P^*AB^*)^{-1}$ and P^*. Hence

$$\|F_6\|_\infty \le \varepsilon\|B^*\|_\infty\|(P^*AB^*)^{-1}\|_\infty\|P^*\|_\infty \ . \tag{4.56}$$

Since $A^+ = B^*(P^*AB^*)^{-1}P^*$, it follows from (4.55) that

$$\begin{aligned} \|A_E^+ - A_C^+\|_\infty \le\ & \|B^*\|_\infty\|F_4\|_\infty\|P^*\|_\infty \\ & + \|F_2\|_\infty\|(P^*AB^*)^{-1}\|_\infty\|P^*\|_\infty \\ & + \|F_5\|_\infty\|P^*\|_\infty + \|F_6\|_\infty \ , \end{aligned} \tag{4.57}$$

where A_E^+ and A_C^+ denote the exact and computed values of A^+, respectively. By (4.54) and (4.56),

$$\|F_5\|_\infty\|P^*\|_\infty + \|F_6\|_\infty \le 2\varepsilon\|B^*\|_\infty\|(P^*AB^*)^{-1}\|_\infty\|P^*\|_\infty \ . \tag{4.58}$$

The result of substituting (4.50) and (4.58) into (4.57) is stated in the following theorem.

Theorem 4.5 *If the Pseudoinverse A^+ of an arbitrary matrix A is computed using Willner's algorithm (see Section 4.1), then the round-off error in the computed value of A^+ is bounded by*

$$\|A_E^+ - A_C^+\|_\infty \le \|P^*\|_\infty\Big[\|B^*\|_\infty\|F_4\|_\infty + \|(P^*AB^*)^{-1}\|_\infty(\|F_2\|_\infty + 2\varepsilon\|B^*\|_\infty)\Big] \ , \tag{4.59}$$

where F_2 and F_4 are defined by (4.26) and (4.40), respectively, and they are bounded by (4.27) and (4.50), respectively.

$\|B^*\|_\infty$, $\|P^*\|_\infty$ and $\|(P^*AB^*)^{-1}\|_\infty$ must be computed in the process of calculating A^+. Our computer program for the computation of the Pseudoinverse includes the evaluation of the right-hand side of (4.59) (see Chapter V).

CHAPTER V

NUMERICAL RESULTS

The computation of a Reflexive Generalized Inverse and the Pseudodinverse was performed using the IBM System 360 FORTRAN IV source language and the IBM System 360/67 computer system at the University of Alberta. Several test matrices, for which the exact Reflexive Generalized Inverse and Pseudoinverse are known, were used. The programs evaluate the norm bounds for the round-off error as given in Chapters III and IV, and a comparison is made between the actual and predicted errors.

5.1 Test Data

5.1.1 Nonsingular Matrices The test data for both the computation of a Reflexive Generalized Inverse and the Pseudoinverse are based on a set of nonsingular matrices A_i of order n with known inverses. For any positive integer k, rectangular matrices of dimension n-by-kn with known Reflexive Generalized Inverses and Pseudoinverses can be constructed by catenating k such identical nonsingular matrices. The following nonsingular matrices denoted A_1, A_2, A_3 and A_4 are due to Newman and Todd [12], while matrices A_5 and A_6 are due to Charmonman and Julius [2].

The matrix A_1 is defined by

$$a_{ij} = \left[\frac{2}{n+1}\right]^{1/2} \sin\left[\frac{ij\pi}{n+1}\right] . \tag{5.1}$$

Since A_1 is symmetric and orthogonal, we have

$$A_1^{-1} = A_1 . \tag{5.2}$$

The matrix A_2 is defined by

$$a_{ij} = \begin{cases} i/j , & i \le j ; \\ j/i , & i > j , \end{cases} \tag{5.3}$$

and the inverse of A_2 is a symmetric, triple-diagonal matrix with

$$a_{ij} = \begin{cases} \dfrac{4i^3}{4i^2-1} , & i=j, \quad i < n ; \\ \dfrac{n^2}{2n-1} , & i=j=n; \\ \dfrac{-i(i+1)}{2i+1} , & |i-j|=1 ; \\ 0 , & |i-j|>1 . \end{cases} \tag{5.4}$$

The matrix A_3 is a triple-diagonal matrix defined by

$$a_{ij} = \begin{cases} -2 , & i=j ; \\ 1 , & |i-j|=1 ; \\ 0 , & |i-j|>1 . \end{cases} \tag{5.5}$$

Its inverse is given by

$$a_{ij} = \begin{cases} \dfrac{-i(n-j+1)}{n+1} , & i \le j ; \\ a_{ji} , & i > j . \end{cases} \tag{5.6}$$

The matrix A_4 is defined by

$$a_{ij} = 2 \min (i,j) - 1 , \tag{5.7}$$

and its inverse is a triple-diagonal matrix given by

$$a_{ij} = \begin{cases} 1.5 , & i=j=1 ; \\ 0.5 , & i=j=n ; \\ 1 , & 1<i=j<n ; \\ -0.5 , & |i-j|=1 ; \\ 0 , & |i-j|>1 . \end{cases} \tag{5.8}$$

The following definitions are used to define the matrices A_5 and A_6.

Definition 5.1 *An r-row circulant is a square matrix of order n in which the i-th row, $i=2,3,\ldots,n$, is obtained from the $(i-1)$-th row by cyclically shifting each element r places to the right.*

Definition 5.2 *An r-column circulant is a square matrix of order n in which the i-th column, $i=2,3,\ldots,n$, is obtained from the $(i-1)$-th column by cyclically shifting each element r places down.*

The matrix A_5 is the r-row circulant with first row $(a,ah,\ldots,ah^{n-1})$, where a and h are arbitrary constants. The inverse of A_5 is the r-column circulant with first column $(b,0,0,\ldots,0,-hb)'$, where

$$b = \frac{1}{a(1-h^n)} \quad . \tag{5.9}$$

The matrix A_6 is the r-row circulant with first row $[a,a+h,\ldots,a+(n-1)h]$. The inverse of A_6 is the r-column circulant with first column $(b-\alpha,b,b,\ldots,b,b+\alpha)'$, where

$$b = \frac{2}{n^2[2a+(n-1)h]} \tag{5.10}$$

and

$$\alpha = \frac{1}{nh} . \tag{5.11}$$

5.1.2 Rectangular Matrices with Known Reflexive Generalized Inverses Let B be a matrix of dimension m-by-n, $m \leq n$, with a nonsingular matrix A_i of order m in its first m columns, i.e.

$$B = [A_i | C] , \tag{5.12}$$

where C is an arbitrary matrix of dimension m-by-(n-m). It can easily be shown that a Reflexive Generalized Inverse of B is

$$G = \begin{bmatrix} A_i^{-1} \\ \hline O \end{bmatrix}, \tag{5.13}$$

where G is of dimension n-by-m. Furthermore, except for round-off error, this is the Reflexive Generalized Inverse that is calculated by the elimination method given in Section 3.1, provided that each pivotal element is chosen only from the first m columns of B. This latter condition insures that rows $m+1, m+2, \ldots, n$ of the matrix

Q defined by (3.9) are unit vectors. Since rows $m+1,m+2,\ldots,n$ of $(\Delta^{-})'$ are zero, it follows that rows $m+1,m+2,\ldots,n$ of the product $Q(\Delta^{-})'$ must be zero. This guarantees that rows $m+1,m+2,\ldots,n$ of G are zero, and therefore, G must be of the form given by (5.13) to satisfy equations (1.8) and (1.9).

Complete pivoting is used in the Gaussian forward elimination for the reduction of the matrix B in (5.12). At the i-th stage in the reduction, the pivotal element is chosen from rows $i,i+1,\ldots,m$ and columns $i,i+1,\ldots,n$. The condition that the pivot be chosen only from columns $i,i+1,\ldots,m$ will be satisfied if the column vectors of C are equal to column vectors of A_i. The data for our program has been constructed by setting C equal to A_i or $[A_i|A_i]$, a matrix of dimension n-by-2n. Therefore, B is of dimension n-by-2n or n-by-3n, and takes the form

(5.14) $$[A_i|A_i]$$

or

(5.15) $$[A_i|A_i|A_i] \quad ,$$

where A_i is one of the six nonsingular matrices of order n defined in Subsection 5.1.1. Then the Reflexive Generalized Inverse obtained by using the algorithm of Section 3.1 is

$$\left[\begin{array}{c} A_i^{-1} \\ \hline 0 \end{array}\right] \tag{5.16}$$

for the matrix (5.14), and

$$\left[\begin{array}{c} A_i^{-1} \\ \hline 0 \\ \hline 0 \end{array}\right] \tag{5.17}$$

for the matrix (5.15). The dimensions of the matrices (5.16) and (5.17) are $2n$-by-n and $3n$-by-n, respectively. Various values of n between 3 and 20 were used for computational purposes.

5.1.3 Rectangular Matrices with Known Pseudoinverses

Let B be a matrix of dimension m-by-km, constructed by placing k identical nonsingular matrices A_i, of order m, side-by-side; i.e.

$$B = [A_i | A_i | A_i | \ldots | A_i] \ . \tag{5.18}$$

Let C be a matrix of dimension km-by-m, constructed by placing k matrices cA_i^{-1} together, where c is a constant; i.e.

$$C = c \begin{bmatrix} A_i^{-1} \\ \hline A_i^{-1} \\ \hline A_i^{-1} \\ \hline \vdots \\ \hline A_i^{-1} \end{bmatrix} . \tag{5.19}$$

Then

$$BC = kcI . \tag{5.20}$$

Therefore,

$$BCB = kcB \tag{5.21}$$

and

$$CBC = kcC . \tag{5.22}$$

For C to be the Pseudoinverse of B, equations (1.1) and (1.2) must be satisfied, i.e.

(5.23) $$BCB = B$$

and

(5.24) $$CBC = C \ .$$

From (5.21) and (5.22), it follows that

(5.25) $$c = \frac{1}{k} \ .$$

Since

(5.26) $$CB = \frac{1}{k} \begin{bmatrix} I & I & \cdots & I \\ \hline I & I & \cdots & I \\ \hline \vdots & \vdots & & \vdots \\ \hline I & I & \cdots & I \end{bmatrix} ,$$

where each identity matrix is of order m, (5.20) and (5.26) verify that equations (1.4) and (1.3) are satisfied. Therefore, C is the Pseudoinverse of B.

The data for our program have been constructed by choosing A_i to be one of the six nonsingular matrices defined in Subsection 5.1.1. Only values of 2 and 3 have been used for k, with n assuming various values between 3 and 20.

5.2 Summary of the Results

The Reflexive Generalized Inverse given by (5.16) or (5.17) and the Pseudoinverse given by (5.19) and (5.25) have been computed, choosing A_i to be one of the six nonsingular matrices of Subsection 5.1.1. The matrix A_5 was generated for the values a=1, h=1.5 and r=1, and a=1, h=2 and r=1 for the computation of the Pseudoinverse. For the computation of a Reflexive Generalized Inverse, the values a=1, h=2 and r=2 or a=1, h=2 and r=3, and the values a=1, h=3 and r=4 or a=1, h=3 and r=3 were used to generate A_5. The second of each pair of values given is necessary for matrices A_5 of certain dimensions, since the first set of values generates a matrix of rank less than n. The matrix A_6 was generated for the values a=1, h=2 and r=1, and a=1, h=3 and r=1 for the computation of the Pseudoinverse. For computation of a Reflexive Generalized Inverse, the values a=1, h=1 and r=2 or a=1, h=1 and r=3, and a=1, h=3 and r=4 or a=1, h=3 and r=3 were used. Eight different nonsingular matrices were generated for the construction of rectangular matrices of dimension 3-by-6, 3-by-9, 5-by-10, 5-by-15, 8-by-16, 8-by-24, 10-by-20, 10-by-30, 15-by-30, 15-by-45 and 20-by-40.

5.2.1 Results for the Reflexive Generalized Inverse

Table 5.1 gives the predicted relative error and the actual relative error incurred in the computation of the Reflexive Generalized Inverse G, given by (5.16) or (5.17), of a matrix A of dimension m-by-2m or m-by-3m. The predicted absolute error, $\|G_E-G_C\|_\infty$, is given by (3.74), and the predicted relative error is defined to be

$$\frac{\|G_E-G_C\|_\infty}{\|G_E\|_\infty} \quad . \tag{5.27}$$

The actual relative error can be computed since the exact Reflexive Generalized Inverse is known.

The first column of Table 5.1 lists the nonsingular matrices A_i of Subsection 5.1.1 used to construct the matrices A, which take either the form of (5.14) or (5.15). The three parenthesized numbers after A_5 and A_6 are, respectively, the values of a, h and r used to construct these matrices. The second column of the table gives the dimension of A. The notation $aE\pm b$ denotes the value $a\times 10^{\pm b}$.

TABLE 5.1

ERROR IN THE REFLEXIVE GENERALIZED INVERSE

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
A_1	3-by-6	0.7344E-4	0.1641E-5
A_2	3-by-6	0.2532E-4	0.7196E-6
A_3	3-by-6	0.4492E-4	0.0
A_4	3-by-6	0.4475E-4	0.5811E-7
$A_5(1,2,2)$	3-by-6	0.8338E-3	0.0
$A_5(1,3,4)$	3-by-6	0.2739E-1	0.1652E-6
$A_6(1,1,2)$	3-by-6	0.2941E-3	0.4952E-6
$A_6(1,3,4)$	3-by-6	0.1409E-1	0.2682E-6
A_1	3-by-9	0.1075E-3	0.1641E-5
A_2	3-by-9	0.3915E-4	0.7196E-6
A_3	3-by-9	0.6689E-4	0.0
A_4	3-by-9	0.6698E-4	0.5811E-7
$A_5(1,2,2)$	3-by-9	0.1171E-2	0.0
$A_5(1,3,4)$	3-by-9	0.3860E-1	0.1652E-6
$A_6(1,1,2)$	3-by-9	0.4038E-3	0.4952E-6
$A_6(1,3,4)$	3-by-9	0.1923E-1	0.2682E-6
A_1	5-by-10	0.7144E-3	0.9780E-6
A_2	5-by-10	0.2189E-3	0.4631E-6
A_3	5-by-10	0.1986E-3	0.3311E-6
A_4	5-by-10	0.4468E-3	0.1025E-5

TABLE 5.1 (Continued)

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
$A_5(1,2,2)$	5-by-10	0.5990	0.0
$A_5(1,3,4)$	5-by-10	0.3711E+3	0.5297E-6
$A_6(1,1,2)$	5-by-10	0.8060E-2	0.4572E-6
$A_6(1,3,4)$	5-by-10	0.5239	0.4951E-6
A_1	5-by-15	0.1086E-2	0.9780E-6
A_2	5-by-15	0.3464E-3	0.4631E-6
A_3	5-by-15	0.3131E-3	0.3311E-6
A_4	5-by-15	0.6786E-3	0.1025E-5
$A_5(1,2,2)$	5-by-15	0.8273	0.0
$A_5(1,3,4)$	5-by-15	0.5301E+3	0.5297E-6
$A_6(1,1,2)$	5-by-15	0.1016E-1	0.4572E-6
$A_6(1,3,4)$	5-by-15	0.6635	0.4951E-6
A_1	8-by-16	0.1494E-1	0.4615E-5
A_2	8-by-16	0.3796E-2	0.1197E-5
A_3	8-by-16	0.2662E-2	0.3636E-6
A_4	8-by-16	0.6602E-2	0.3008E-5
$A_5(1,2,3)$	8-by-16	0.4468E+4	0.0
$A_5(1,3,3)$	8-by-16	0.1137E+8	0.6392E-6
$A_6(1,1,3)$	8-by-16	0.1329	0.7729E-6
$A_6(1,3,3)$	8-by-16	0.5950E+1	0.3749E-5
A_1	8-by-24	0.2484E-1	0.4615E-5

TABLE 5.1 (Continued)

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
A_2	8-by-24	0.6180E-2	0.1197E-5
A_3	8-by-24	0.4603E-2	0.3636E-6
A_4	8-by-24	0.1037E-1	0.3008E-5
$A_5(1,2,3)$	8-by-24	0.6076E+4	0.0
$A_5(1,3,3)$	8-by-24	0.1602E+8	0.6392E-6
$A_6(1,1,3)$	8-by-24	0.1608	0.7729E-6
$A_6(1,3,3)$	8-by-24	0.7907E+1	0.3749E-5
A_1	10-by-20	0.9892E-1	0.8823E-5
A_2	10-by-20	0.2221E-1	0.1199E-5
A_3	10-by-20	0.1648E-1	0.3854E-6
A_4	10-by-20	0.3559E-1	0.2895E-5
$A_5(1,2,3)$	10-by-20	0.4699E+6	0.0
$A_5(1,3,3)$	10-by-20	0.9169E+9	0.7963E-6
$A_6(1,1,3)$	10-by-20	0.3535	0.7900E-6
$A_6(1,3,3)$	10-by-20	0.2970E+2	0.3735E-5
A_1	10-by-30	0.1713	0.8823E-5
A_2	10-by-30	0.3665E-1	0.1199E-5
A_3	10-by-30	0.2951E-1	0.3854E-6
A_4	10-by-30	0.5765E-1	0.2895E-5
$A_5(1,2,3)$	10-by-30	0.6540E+6	0.0
$A_5(1,3,3)$	10-by-30	0.1310E+10	0.7963E-6

TABLE 5.1 (Continued)

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
$A_6(1,1,3)$	10-by-30	0.4532	0.7900E-6
$A_6(1,3,3)$	10-by-30	0.3620E+2	0.3735E-5
A_1	15-by-30	0.8526E+1	0.4655E-5
A_2	15-by-30	0.1502E+1	0.5494E-5
A_3	15-by-30	0.1069E+1	0.0
A_4	15-by-30	0.2373E+1	0.7918E-5
$A_5(1,2,2)$	15-by-30	0.1108E+10	0.0
$A_5(1,3,4)$	15-by-30	0.9105E+14	0.1304E-5
$A_6(1,1,2)$	15-by-30	0.5697E+1	0.1931E-5
$A_6(1,3,4)$	15-by-30	0.4282E+3	0.5759E-5
A_1	15-by-45	0.1554E+2	0.4655E-5
A_2	15-by-45	0.2586E+1	0.5494E-5
A_3	15-by-45	0.1999E+1	0.0
A_4	15-by-45	0.4002E+1	0.7918E-5
$A_5(1,2,2)$	15-by-45	0.1686E+10	0.0
$A_5(1,3,4)$	15-by-45	0.1411E+15	0.1304E-5
$A_6(1,1,2)$	15-by-45	0.8284E+1	0.1931E-5
$A_6(1,3,4)$	15-by-45	0.5404E+3	0.5759E-5
A_1	20-by-40	0.1029E+4	0.7707E-5
A_2	20-by-40	0.3248E+3	0.1008E-4
A_3	20-by-40	0.9273E+2	0.2363E-6

TABLE 5.1 (Continued)

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
A_4	20-by-40	0.4909E+3	0.1090E-4
$A_5(1,2,3)$	20-by-40	0.1114E+15	0.0
$A_5(1,3,3)$	20-by-40	0.1398E+21	0.1232E+1
$A_6(1,1,3)$	20-by-40	0.1366E+4	0.1120E-4
$A_6(1,3,3)$	20-by-40	0.7873E+5	0.5103E-5

5.2.2 Results for the Pseudoinverse Table 5.2 gives the predicted relative error and the actual relative error incurred in the computation of the Pseudoinverse A^+, given by (5.19) and (5.25), of a matrix A of dimension m-by-2m or m-by-3m. The predicted absolute error, $\|A_E^+ - A_C^+\|_\infty$, is given by (4.59), and the predicted relative error is defined to be

$$\frac{\|A_E^+ - A_C^+\|_\infty}{\|A_E^+\|_\infty} \quad . \tag{5.28}$$

The actual relative error can be computed since the exact Pseudoinverse is known.

The first column of Table 5.2 lists the nonsingular matrices A_i of Subsection 5.1.1 used to construct the matrices A, which take either the form of (5.14) or (5.15). The three parenthesized numbers after A_5 and A_6 are, respectively, the values of a, h and r used to construct these matrices. The second column of the table gives the dimension of A. In many cases, the condition specified by (4.47) does not hold, and no predicted relative error can be computed for these matrices.

TABLE 5.2

ERROR IN THE PSEUDOINVERSE

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
A_1	3-by-6	0.8974E-4	0.2514E-5
A_2	3-by-6	0.1387E-1	0.4023E-5
A_3	3-by-6	0.4948E-2	0.1252E-5
A_4	3-by-6	0.1105	0.3219E-5
$A_5(1,1.5,1)$	3-by-6	0.1595E-2	0.5861E-5
$A_5(1,2,1)$	3-by-6	0.3203E-3	0.6606E-6
$A_6(1,2,1)$	3-by-6	0.2692E-3	0.9656E-6
$A_6(1,3,1)$	3-by-6	0.1928E-3	0.1460E-5
A_1	3-by-9	0.8974E-4	0.2095E-5
A_2	3-by-9	0.1387E-1	0.8404E-5
A_3	3-by-9	0.4948E-2	0.8762E-5
A_4	3-by-9	0.1105	0.1878E-5
$A_5(1,1.5,1)$	3-by-9	0.1595E-2	0.3100E-5
$A_5(1,2,1)$	3-by-9	0.3203E-3	0.4433E-6
$A_6(1,2,1)$	3-by-9	0.2692E-3	0.7544E-6
$A_6(1,3,1)$	3-by-9	0.1928E-3	0.9835E-6
A_1	5-by-10	0.5797E-2	0.2490E-5
A_2	5-by-10	---	0.3100E-4
A_3	5-by-10	0.4170	0.9749E-5
A_4	5-by-10	---	0.7093E-5

TABLE 5.2 (Continued)

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
$A_5(1,1.5,1)$	5-by-10	0.4170E-2	0.1171E-5
$A_5(1,2,1)$	5-by-10	0.9009E-3	0.1008E-5
$A_6(1,2,1)$	5-by-10	0.4285E-2	0.3193E-5
$A_6(1,3,1)$	5-by-10	0.3244E-2	0.2076E-5
A_1	5-by-15	0.5797E-2	0.2075E-5
A_2	5-by-15	---	0.1886E-4
A_3	5-by-15	0.4171	0.9521E-4
A_4	5-by-15	---	0.1010E-4
$A_5(1,1.5,1)$	5-by-15	0.4170E-2	0.1794E-5
$A_5(1,2,1)$	5-by-15	0.9009E-3	0.8430E-6
$A_6(1,2,1)$	5-by-15	0.4285E-2	0.2638E-5
$A_6(1,3,1)$	5-by-15	0.3244E-2	0.2103E-5
A_1	8-by-16	---	0.6243E-5
A_2	8-by-16	---	0.1755E-3
A_3	8-by-16	---	0.5575E-4
A_4	8-by-16	---	0.1529E-4
$A_5(1,1.5,1)$	8-by-16	0.1814E-1	0.4313E-5
$A_5(1,2,1)$	8-by-16	0.2918E-2	0.1650E-5
$A_6(1,2,1)$	8-by-16	0.1125E+1	0.5289E-5
$A_6(1,3,1)$	8-by-16	0.7700	0.1771E-5
A_1	8-by-24	---	0.4962E-5
A_2	8-by-24	---	0.1630E-3

TABLE 5.2 (Continued)

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
A_3	8-by-24	---	0.6708E-3
A_4	8-by-24	---	0.2341E-4
$A_5(1,1.5,1)$	8-by-24	0.1814E-1	0.3664E-5
$A_5(1,2,1)$	8-by-24	0.2918E-2	0.2239E-5
$A_6(1,2,1)$	8-by-24	0.1125E+1	0.1757E-5
$A_6(1,3,1)$	8-by-24	0.7700	0.7643E-5
A_1	10-by-20	---	0.1233E-4
A_2	10-by-20	---	0.2342E-3
A_3	10-by-20	---	0.1702E-3
A_4	10-by-20	---	0.4673E-4
$A_5(1,1.5,1)$	10-by-20	0.5584E-1	0.7403E-5
$A_5(1,2,1)$	10-by-20	0.5352E-2	0.2238E-5
$A_6(1,2,1)$	10-by-20	---	0.2979E-5
$A_6(1,3,1)$	10-by-20	---	0.1031E-4
A_1	10-by-30	---	0.1078E-4
A_2	10-by-30	---	0.1594E-3
A_3	10-by-30	---	0.1699E-2
A_4	10-by-30	---	0.6107E-4
$A_5(1,1.5,1)$	10-by-30	0.5584E-1	0.7349E-5
$A_5(1,2,1)$	10-by-30	0.5352E-2	0.2367E-5
$A_6(1,2,1)$	10-by-30	---	0.2339E-4
$A_6(1,3,1)$	10-by-30	---	0.1283E-4

TABLE 5.2 (Continued)

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
A_1	15-by-30	---	0.8180E-5
A_2	15-by-30	---	0.1050E-2
A_3	15-by-30	---	0.1160E-2
A_4	15-by-30	---	0.9946E-3
$A_5(1,1.5,1)$	15-by-30	0.2512E+1	0.5378E-5
$A_5(1,2,1)$	15-by-30	0.1685E-1	0.7805E-5
$A_6(1,2,1)$	15-by-30	---	0.2690E-5
$A_6(1,3,1)$	15-by-30	---	0.8783E-5
A_1	15-by-45	---	0.6357E-5
A_2	15-by-45	---	0.1738E-2
A_3	15-by-45	---	0.1114E-1
A_4	15-by-45	---	0.4640E-3
$A_5(1,1.5,1)$	15-by-45	0.2512E+1	0.4315E-5
$A_5(1,2,1)$	15-by-45	0.1685E-1	0.5256E-5
$A_6(1,2,1)$	15-by-45	---	0.1582E-4
$A_6(1,3,1)$	15-by-45	---	0.1282E-4
A_1	20-by-40	---	0.1536E-4
A_2	20-by-40	---	0.8582E-2
A_3	20-by-40	---	0.3796E-2
A_4	20-by-40	---	0.1100E-2
$A_5(1,1.5,1)$	20-by-40	---	0.2162E-4

TABLE 5.2 (Continued)

MATRIX	DIMENSION	PREDICTED RELATIVE ERROR	ACTUAL RELATIVE ERROR
$A_5(1,2,1)$	20-by-40	0.3908E-1	0.3918E-5
$A_6(1,2,1)$	20-by-40	---	0.2529E-4
$A_6(1,3,1)$	20-by-40	---	0.1741E-4

CHAPTER VI

CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

6.1 Conclusions

The following conclusions are drawn from the numerical results:

1. The predicted round-off error unfortunately is substantially larger than the actual round-off error in most cases. This is due to the fact that in the error analysis, we allow for the maximum amount of round-off error in each calculation. In practice, the round-off error incurred in a computation is seldom the maximum possible. It is important, however, to realize that matrices can be constructed such that the actual total round-off error approaches the predicted value.

2. The comparatively large values obtained for the predicted relative error in computing the Reflexive Generalized Inverse of a matrix constructed from A_5 are due to large values of the bound on $\|F_2\|_\infty$. Consequently, the computation of $(\Delta^-)'$ should be done in multiple precision arithmetic to lower the bound on $\|F_2\|_\infty$, and thus improve the bound for the Reflexive Generalized Inverse.

3. The actual relative error is, on the average, smaller for the computed Reflexive Generalized Inverse, indicating that its computation is less susceptible to the accumulation of round-off errors than is the computation of the Pseudoinverse. This remark, however, applies only to the two algorithms that we have programmed, and is based solely on the set of data tested.

4. The complete listing of the numerical results shows that the bound given by (4.59) is approximately equal to the product of $\|P^*\|_\infty$ and $\|F_4\|_\infty$. Hence, these are the only two values which need be computed to obtain an approximate bound on the round-off error in the computed Pseudoinverse.

6.2 Suggestions for Future Research

1. In some applications of generalized inverses, it is irrelevant which of the four generalized inverses defined in Chapter I is computed. Hence, error analyses of other algorithms could suggest which of the four generalized inverses, and, in particular, which algorithms, are the least susceptible to the accumulation of round-off error.

2. The error bound for the matrix B^*, $\|F_2\|_\infty$, in Willner's Algorithm would be significantly smaller for large matrices if the ∞-norm replaced the 1-norm in (4.27).

This would require use of the 1-norm, instead of the ∞-norm, in bounding the error for the computed Hermite normal form. The error bound for the computed Pseudoinverse may then be significantly smaller, because, from observation of the numerical results, $\|F\|_\infty$, which is a factor in the bound for $\|F_4\|_\infty$, is approximately equal to $\|P^*\|_\infty \|A\|_\infty \|F_2\|_\infty$. As stated in the fourth conclusion of Section 6.1, $\|F_4\|_\infty$ is one of the two major factors of the bound on the computed Pseudoinverse.

3. A major deficiency in the bound given by (4.59) for the round-off error incurred in computing the Pseudoinverse is the requirement that condition (4.47) hold. This condition might be relaxed if a simple relationship could be found between the exact and computed values of $\|(PAB+F)^{-1}\|_\infty$. Perhaps, for example, it could be shown that

$$\|(PAB+F)_E^{-1}\|_\infty \leq \|(PAB+F)_C^{-1}\|_\infty,$$

provided the condition number of $(PAB+F)$ is greater than or equal to one.

4. The predicted relative error for the computed Reflexive Generalized Inverse varies greatly with different test matrices of the same dimension, although the actual

relative error is quite stable. Perhaps large values of the predicted relative error could be related to a condition number or a norm of the matrix. This would give a criterion for deciding whether or not the predicted relative error is a reasonable approximation to the actual relative error.

BIBLIOGRAPHY

1. Ben-Israel, A. and Cohen, D., "On Iterative Computation of Generalized Inverses and Associated Projections", J. Soc. Indust. Appl. Math. Ser. Numer. Anal., 3, 1966, pp. 410-419.

2. Charmonman, S. and Julius, R.S., "Explicit Inverses and Condition Numbers of Certain Circulants", Math. Comp. (To appear April 1968).

3. Fox L., An Introduction to Numerical Linear Algebra, Clarendon Press, Oxford, 1964.

4. Goldman, A.J. and Zelen, M., "Weak Generalized Inverses and Minimum Variance Linear Unbiased Estimation", J. Res. National Bureau of Standards, 68B, 1964, pp. 151-172.

5. Graybill, F.A., Meyer, C.D., and Painter, R.J., "Note on the Computation of the Generalized Inverse of a Matrix", SIAM Rev., 8, 1966, pp. 522-524.

6. Greville, T.N.E., "The Pseudoinverse of a Rectangular or Singular Matrix and its application to the Solution of Systems of Linear Equations", SIAM Rev., 1, 1959, pp. 38-43.

7. Greville, T.N.E., "Some Applications of the Pseudoinverse of a Matrix", SIAM Rev., 2, 1960, pp. 15-22.

8. Householder, A.S., "The Approximate Solution of Matrix Problems", J. Assoc. Comput. Mach., 5, 1958, pp. 205-243.

9. Korganoff, A. and Pavel-Parvu, M., Éléments de Théorie des Matrices Carrées et Rectangles en Analyse Numérique, Dunod, Paris, 1967.

10. Marcus, M. and Minc, H., A Survey of Matrix Theory and Matrix Inequalities, Allyn and Bacon, Boston, 1964.

11. Moore, E.H., "General Analysis", Part I, Mem. Amer. Philos. Soc., 1, 1935.

12. Newman, M. and Todd, J., "The Evaluation of Matrix Inversion Programs", J. Soc. Indust. Appl. Math., 6, 1958, pp. 466-476.

13. Penrose, R., "A Generalized Inverse for Matrices", Proc. Cambridge Philos. Soc., 51, 1955, pp. 406-413.

14. Pyle, L.D., "Generalized Inverse Computations Using the Gradient Projection Method", J. Assoc. Comput. Mach., 11, 1964, pp. 422-428.

15. Rado, R., "Note on Generalized Inverses of Matrices", Proc. Cambridge Philos. Soc., 52, 1956, pp. 600-601.

16. Rao, C.R., "A Note on a Generalized Inverse of a Matrix with Applications to Problems in Mathematical Statistics", J. Roy. Stat. Soc. (B), 24, 1962, pp. 152-158.

17. Rohde, C.A., Contributions to the Theory, Computation and Application of Generalized Inverses, Ph.D. Thesis, University of North Carolina at Raleigh, 1964.

18. Rust, B., Burrus, W.R., and Schneeberger, C., "A Simple Algorithm for Computing the Generalized Inverse of a Matrix", Comm. Assoc. Comput. Mach., 9, 1966, pp. 381-385, 387.

19. Wang, P., Numerical and Matrix Methods in Structural Mechanics, Wiley and Sons, New York, 1966.

20. Wilkinson, J.H., "Error Analysis of Direct Methods of Matrix Inversion", J. Assoc. Comput. Mach., 8, 1961, pp. 281-330.

21. Wilkinson, J.H., Rounding Errors in Algebraic Processes, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1963.

22. Wilkinson, J.H., The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

23. Willner, L.B., "An Elimination Method for Computing the Generalized Inverse", Math. Comp.,21, 1967, pp. 227-229.

APPENDIX

LISTINGS OF FORTRAN IV SUBPROGRAMS

The following pages contain listings of the IBM System 360 FORTRAN IV subprograms used to compute a Reflexive Generalized Inverse and the Pseudoinverse of a matrix A. The SUBROUTINE AGI computes a Reflexive Generalized Inverse, and the SUBROUTINE PSEUDO computes the Pseudoinverse. The SUBROUTINE NORM computes the ∞-norm of an arbitrary matrix. The FUNCTION subprogram AMAX computes the element of maximum modulus in a certain submatrix of an arbitrary matrix.

The unit round-off error for the IBM System 360/67 computer is 2^{-21}. Although the word length is 32 bits, one bit is used for the sign, seven bits are used for the exponent, and the three leading bits of the fractional part may be zero, as normalization is performed in hexadecimal.

The matrices of round-off errors are referenced in comment statements by the names given them in Chapters III and IV.

```
      SUBROUTINE AGI(M,N,ZERO,A,Q,GNORM,RELERR)
      DIMENSION A(50,50),U(50,50),P(50,50),PD(50,50),
     1 Q(50,50),NR(50),NC(50),DEL(50)
      DOUBLE PRECISION E,EPSI,TEMP,F2NORM,E1,E2,DP,DM(51)
C
C THIS SUBROUTINE CALCULATES A REFLEXIVE GENERALIZED
C INVERSE OF A MATRIX A OF DIMENSION M-BY-N AND
C CALCULATES AN UPPER BOUND FOR THE ROUND-OFF ERROR
C INCURRED IN THE CALCULATION.  THE REFLEXIVE GENERALIZED
C INVERSE IS RETURNED VIA THE ARRAY Q.
C GNORM IS THE VALUE OF THE INFINITY NORM OF THE EXACT
C REFLEXIVE GENERALIZED INVERSE OF A.  IT IS USED TO
C COMPUTE THE PREDICTED RELATIVE ERROR, RELERR, IN THE
C COMPUTED REFLEXIVE GENERALIZED INVERSE.
C EPSI IS THE UNIT ROUND-OFF ERROR FOR THE IBM 360/67.
C ZERO IS THE ZERO CRITERION USED IN THE FORWARD
C ELIMINATION.
C
      EPSI=2.D0**(-21)
C
C FORM A UNIT MATRIX IN U.
C
      DATA U/2500*0/
      DO 10 I=1,50
 10   U(I,I)=1
C
C NRANK IS THE CALCULATED RANK OF A.  G IS THE VALUE OF
C THE ELEMENT OF MAXIMUM MODULUS IN THE REDUCTION OF A.
C
      MM1=M-1
      MP1=M+1
      MP2=M+2
      NRANK=M
      NFLAG=0
      G=0
C
C THE BEGINNING OF THE FORWARD ELIMINATION.
C
      DO 45 I=1,M
      G=AMAX1(G,AMAX(I,M,N,A))
      IP1=I+1
C
C THE COMPLETE PIVOTING.  THE VECTORS NR AND NC NOTE
C THE NECESSARY ROW AND COLUMN INTERCHANGES RESPECITVELY.
C
      DMAX=0
      NR(I)=I
      NC(I)=I
```

```
      DO 20 J=I,M
      DO 20 K=I,N
      IF(DMAX.GE.ABS(A(J,K))) GOTO 20
      DMAX=ABS(A(J,K))
      NR(I)=J
      NC(I)=K
 20   CONTINUE
      IF(DMAX.LT.ZERO) NRANK=I-1
C
C IF, AT ANY STAGE OF THE ELIMINATION, THE REMAINDER OF
C THE MATRIX IS LESS THAN THE ZERO CRITERION, THEN NO
C FURTHER ELIMINATION IS NECESSARY, AND THE RANK OF A IS
C FOUND TO BE LESS THAN M.
C
      IF(NRANK.NE.M) GOTO 50
C
C INTERCHANGE COLUMNS IF NECESSARY.
C
      IF(NC(I).EQ.I) GOTO 30
      DO 25 K=1,M
      DMAX=A(K,I)
      A(K,I)=A(K,NC(I))
 25   A(K,NC(I))=DMAX
 30   IF(I.EQ.M) GOTO 50
C
C INTERCHANGE ROWS IF NECESSARY.
C
      IF(NR(I).EQ.I) GOTO 36
      DO 35 J=I,N
      DMAX=A(I,J)
      A(I,J)=A(NR(I),J)
 35   A(NR(I),J)=DMAX
C
C DETERMINE WHETHER OR NOT THE REMAINDER OF THE MATRIX
C IS ALL ZEROS.
C
 36   DMAX=0
      DO 40 J=IP1,M
      DO 40 K=I,N
      IF(DMAX.GE.ABS(A(J,K))) GOTO 40
      DMAX=ABS(A(J,K))
 40   CONTINUE
      IF(DMAX.LT.ZERO) NRANK=I
      IF(NRANK.NE.M) GOTO 50
C
C THE REDUCTION OF ROWS (I+1, ..., M) OF A.
C
```

```
      DO 45 J=IP1,M
      DP=-A(J,I)/A(I,I)
      A(J,I)=DP
      DO 45 K=IP1,N
 45   A(J,K)=A(J,K)+DP*A(I,K)
C
C INITIALIZE THE MATRIX P.
C
 50   IF(NRANK.NE.M) GOTO 51
      NRANK=MM1
      NFLAG=1
 51   DO 65 J=1,M
      DO 65 K=1,M
 65   P(J,K)=U(J,K)
      NRP1=NRANK+1
      DO 68 I=NRP1,M
 68   P(I,NRANK)=A(I,NRANK)
C
C CALCULATION OF P.
C
      DO 95 I=1,NRANK
      ID=NRP1-I
C
C INTERCHANGE COLUMNS IF NECESSARY.
C
      IF(NR(ID).EQ.ID) GOTO 71
      DO 70 J=1,M
      DMAX=P(J,ID)
      P(J,ID)=P(J,NR(ID))
 70   P(J,NR(ID))=DMAX
 71   IF(I.EQ.NRANK) GOTO 95
C
C FORM THE MATRIX P(ID-1) IN PD.
C
      DO 75 J=1,M
      DO 75 K=1,M
 75   PD(J,K)=U(J,K)
      DO 80 J=ID,M
 80   PD(J,ID-1)=A(J,ID-1)
C
C MULTIPLY P BY PD.
C
      DO 90 J=1,M
      DO 85 K=1,M
      DM(K)=0
      DO 85 L=1,M
 85   DM(K)=DM(K)+DBLE(P(J,L))*PD(L,K)
      DO 90 K=1,M
 90   P(J,K)=DM(K)
 95   CONTINUE
```

```
C
C THE INFINITY NORM OF P IS STORED IN PNORM.
C
      CALL NORM(PNORM,M,M,P)
C
C CALCULATION OF Q.
C
      NRANK=NRANK+NFLAG
      NRP1=NRANK+1
      NRP2=NRANK+2
      DO 100 I=1,N
      DO 100 J=1,N
  100 Q(I,J)=U(I,J)
      DO 106 I=1,NRANK
      IP1=I+1
      DO 106 J=IP1,NRP1
      L=J-1
      DP=0
      DO 105 K=I,L
  105 DP=DP-DBLE(A(K,J))/A(K,K)*Q(I,K)
  106 Q(I,J)=DP
      IF(N.LE.NRP1) GOTO 112
      DO 111 I=1,NRANK
      DO 111 J=NRP2,N
      DP=0
      DO 110 K=I,NRANK
  110 DP=DP-DBLE(A(K,J))/A(K,K)*Q(I,K)
  111 Q(I,J)=DP
C
C INTERCHANGE THE ROWS OF Q IF NECESSARY.
C
  112 DO 120 I=1,NRANK
      ID=NRP1-I
      IF(NC(ID).EQ.ID) GOTO 120
      DO 115 J=1,N
      DMAX=Q(NC(ID),J)
      Q(NC(ID),J)=Q(ID,J)
  115 Q(ID,J)=DMAX
  120 CONTINUE
C
C THE INFINITY NORM OF Q IS STORED IN QNORM.
C
      CALL NORM(QNORM,N,N,Q)
C
C CALCULATION OF THE MATRIX DELTA-.  ITS "DIAGONAL"
C ELEMENTS ARE STORED IN THE VECTOR DEL.
C
      DO 125 I=1,NRANK
  125 DEL(I)=1/A(I,I)
      IF(NRANK.EQ.M) GOTO 131
```

```
      DO 130 I=NRP1,M
  130 DEL(I)=0
C
C THE INFINITY NORM OF DELTA- IS STORED IN DELNRM.
C
  131 DELNRM=0
      DO 135 I=1,NRANK
  135 DELNRM=AMAX1(DELNRM,ABS(DEL(I)))
C
C THE INFINITY NORM OF THE MATRIX F2 IS STORED IN F2NORM.
C
      F2NORM=0
      DO 140 I=1,NRANK
      E=(I-1)*2.01*G*EPSI
      TEMP=DABS((DEL(I)*EPSI-E)/(DEL(I)*(DEL(I)+E)))
  140 F2NORM=DMAX1(F2NORM,TEMP)
C
C CALCULATION OF THE PREDICTED ABSOLUTE ERROR IN THE
C COMPUTED REFLEXIVE GENERALIZED INVERSE.
C
      E1=NRANK*(N-NRANK+1)*2**(NRANK-2)*EPSI
      E2=DELNRM+F2NORM
      ERROR=(E1*E2+GNORM*F2NORM)*PNORM+(NRANK-1)*
     1 2**NRANK*EPSI*E2*(QNORM+E1)+2*EPSI*QNORM*
     2 DELNRM*PNORM
      RELERR=ERROR/GNORM
C
C MULTIPLICATION OF Q BY DELTA-.
C
      DO 145 I=1,M
      DO 145 J=1,N
  145 Q(J,I)=Q(J,I)*DEL(I)
C
C MULTIPLICATION OF (Q*DELTA-) BY P.  THE REFLEXIVE
C GENERALIZED INVERSE OF A IS STORED IN THE FIRST M
C COLUMNS OF Q.
C
      DO 155 I=1,N
      DO 150 J=1,M
      DM(J)=0
      DO 150 K=1,M
  150 DM(J)=DM(J)+DBLE(Q(I,K))*P(K,J)
      DO 155 J=1,M
  155 Q(I,J)=DM(J)
      RETURN
      END
```

```
      SUBROUTINE PSEUDO(M,N,ZERO,A,G,GNORM,RELERR)
      DIMENSION A(50,50),IDENT(50),AA(50,50),P(50,50),
     1 G(50,50)
      DOUBLE PRECISION DP,DM,EPSI
C
C THIS SUBROUTINE CALCULATES THE PSEUDOINVERSE OF A
C MATRIX A OF DIMENSION M-BY-N AND CALCULATES AN UPPER
C BOUND FOR THE ROUND-OFF ERROR INCURRED IN THE
C COMPUTATIONS.  THE PSEUDOINVERSE IS RETURNED VIA THE
C ARRAY G.
C ZERO IS THE ZERO CRITERION USED IN THE FORWARD
C ELIMINATION.  GNORM IS THE VALUE OF THE INFINITY NORM
C OF THE EXACT PSEUDOINVERSE OF A.  IT IS USED TO COMPUTE
C THE PREDICTED RELATIVE ERROR, RELERR, IN THE COMPUTED
C PSEUDOINVERSE.
C EPSI IS THE UNIT ROUND-OFF ERROR FOR THE IBM 360/67.
C
      EPSI=2.D0**(-21)
C
C THE MATRIX A IS STORED IN AA FOR LATER USE.
C THE INFINITY NORM OF A IS STORED IN ANORM.
C
      DO 10 I=1,M
      DO 10 J=1,N
 10   AA(I,J)=A(I,J)
      CALL NORM(ANORM,M,N,A)
C
C AT EACH STAGE OF THE REDUCTION, NC IS THE NUMBER OF THE
C COLUMN OF A IN WHICH THE PIVOTAL ELEMENT IS LOCATED.
C THUS COLUMN NC OF A WILL BE REDUCED TO A COLUMN VECTOR
C OF THE UNIT MATRIX AT EACH STAGE.
C
      NC=0
C
C THE BEGINNING OF THE CALCULATION OF THE HERMITE NORMAL
C FORM OF A.  F2NORM DENOTES THE INFINITY NORM OF THE
C MATRIX OF ROUND-OFF ERRORS INCURRED IN COMPUTING THE
C HERMITE NORMAL FORM OF A.
C
      F2NORM=1
      DO 70 I=1,M
      NC=NC+1
      IF(NC.GT.N) GOTO 75
      NCP1=NC+1
      IP1=I+1
C
C USE PARTIAL PIVOTING TO FIND THE PIVOTAL ELEMENT FOR
C THE I-TH ROW.
C
      DMAX=ABS(A(I,NC))
```

```
      L=I
      IF(I.EQ.M) GOTO 21
      DO 20 J=IP1,M
      IF(DMAX.GE.ABS(A(J,I))) GOTO 20
      L=J
      DMAX=ABS(A(J,I))
 20   CONTINUE
 21   IF(DMAX.GT.ZERO) GOTO 40
      IF(NC.EQ.N) GOTO 75
C
C IF ALL OF THE POSSIBLE PIVOTAL ELEMENTS IN COLUMN NC
C ARE ZERO, SEARCH FOR A PIVOTAL ELEMENT IN COLUMNS
C (NC+1, ... , N).
C
      DO 30 J=NCP1,N
      NC=NC+1
C
C USE PARTIAL PIVOTING TO LOCATE THE LARGEST POSSIBLE
C PIVOTAL ELEMENT IN COLUMN J.
C
      DMAX=ABS(A(I,J))
      L=I
      IF(I.EQ.M) GOTO 26
      DO 25 K=IP1,M
      IF(DMAX.GE.ABS(A(K,J))) GOTO 25
      L=K
      DMAX=ABS(A(K,J))
 25   CONTINUE
 26   NCP1=NC+1
      IF(DMAX.GT.ZERO) GOTO 40
 30   CONTINUE
C
C AS ALL OF THE POSSIBLE PIVOTS ARE ZERO, THE HERMITE
C NORMAL FORM OF A HAS BEEN FOUND.
C
      GOTO 75
 40   IF(L.EQ.I) GOTO 46
C
C INTERCHANGE ROWS IF NECESSARY.
C
      DO 45 J=NC,N
      DMAX=A(I,J)
      A(I,J)=A(L,J)
 45   A(L,J)=DMAX
 46   IF(NC.EQ.N) GOTO 61
C
C THE REDUCTION OF COLUMNS (NC+1, ... , N) OF A USING
C THE FORWARD ELIMINATION OF GAUSSIAN ELIMINATION WITH
C ACCUMULATION OF INNER PRODUCTS.
C
```

```
      DM=0
      DO 55 J=1,M
      IF(J.EQ.I) GOTO 55
      DP=-A(J,NC)/A(I,NC)
      DM=DM+ABS(DP)
      DO 50 L=NCP1,N
 50   A(J,L)=A(J,L)+DP*A(I,L)
 55   CONTINUE
      DM=DM+1/ABS(A(I,NC))
      F2NORM=F2NORM*DMAX1(1.D0,DM)
C
C DIVIDE THE I-TH ROW OF A BY THE PIVOTAL ELEMENT.
C
      DO 60 J=NCP1,N
 60   A(I,J)=A(I,J)/A(I,NC)
C
C SET COLUMN NC OF A EQUAL TO THE I-TH COLUMN OF THE UNIT
C MATRIX.
C
 61   DO 65 J=1,M
 65   A(J,NC)=0
      A(I,NC)=1
C
C THE VECTOR IDENT INDICATES WHICH COLUMNS OF THE HERMITE
C NORMAL FORM OF A ARE EQUAL TO COLUMNS OF THE UNIT
C MATRIX.  NRANK IS THE RANK OF A.
C
      IDENT(I)=NC
 70   NRANK=I
 75   F2NORM=NRANK*EPSI*F2NORM
C
C CALCULATION OF THE MATRIX P-TRANSPOSE.
C
      DO 80 I=1,NRANK
      J=IDENT(I)
      DO 80 K=1,M
 80   P(I,K)=AA(K,J)
C
C THE INFINITY NORM OF P-TRANSPOSE IS STORED IN PNORM.
C
      CALL NORM(PNORM,NRANK,M,P)
C
C MULTIPLY P-TRANSPOSE BY A USING ACCUMULATION OF
C INNER PRODUCTS.
C
      DO 86 I=1,NRANK
      DO 86 J=1,N
      DP=0
      DO 85 K=1,M
 85   DP=DP+DBLE(P(I,K))*AA(K,J)
```

```
 86   G(I,J)=DP
C
C THE INFINITY NORM OF P-TRANSPOSE * A IS STORED IN
C PANORM.  F1NORM IS THE INFINITY NORM OF THE MATRIX OF
C ROUND-OFF ERRORS INCURRED IN COMPUTING P-TRANSPOSE * A.
C
      CALL NORM(PANORM,NRANK,N,G)
      F1NORM=EPSI*PNORM*ANORM
C
C MULTIPLY (P-TRANSPOSE * A) BY B-TRANSPOSE USING
C ACCUMULATION OF INNER PRODUCTS.
C
      DO 91 I=1,NRANK
      DO 91 J=1,NRANK
      DP=0
      DO 90 K=1,N
 90   DP=DP+DBLE(G(I,K))*A(J,K)
 91   AA(I,J)=DP
C
C INVERT THE MATRIX (P-TRANSPOSE * A * B-TRANSPOSE).
C CALL THE INVERTED MATRIX Y.
C GG DENOTES THE ELEMENT OF MAXIMUM MODULUS IN THE
C COMPUTATION OF Y.
C
      CALL AINV(AA,G,NRANK,GG)
C
C THE INFINITY NORM OF Y IS STORED IN PABNRM.
C
      CALL NORM(PABNRM,NRANK,NRANK,G)
C
C MULTIPLY B-TRANSPOSE BY Y USING ACCUMULATION OF
C INNER PRODUCTS.
C
      DO 96 I=1,N
      DO 96 J=1,NRANK
      DP=0
      DO 95 K=1,NRANK
 95   DP=DP+DBLE(A(K,I))*G(K,J)
 96   AA(I,J)=DP
C
C MULTIPLY (B-TRANSPOSE * Y) BY P-TRANSPOSE USING
C ACCUMULATION OF INNER PRODUCTS.  THE RESULT, THE
C PSEUDOINVERSE OF A, IS STORED IN G.
C
      DO 99 I=1,N
      DO 99 J=1,M
      DP=0
      DO 98 K=1,NRANK
 98   DP=DP+DBLE(AA(I,K))*P(K,J)
 99   G(I,J)=DP
```

```
C
C THE MATRIX B-TRANSPOSE IS STORED IN P.
C
      DO 100 I=1,NRANK
      DO 100 J=1,N
  100 P(J,I)=A(I,J)
C
C THE INFINITY NORMS OF THE MATRICES B-TRANSPOSE, F3, F,
C INVERSE OF (P-TRANSPOSE * A * B-TRANSPOSE +F), AND F4
C ARE STORED IN BNORM, F3NORM, FNORM, PABFNM, AND F4NORM
C RESPECTIVELY. C AND D ARE CONSTANTS.  THE PREDICTED
C VALUE OF THE ABSOLUTE ROUND-OFF ERROR IS STORED IN
C ERROR.  THE PREDICTED RELATIVE ERROR IS STORED IN
C RELERR.
C
      CALL NORM(BNORM,N,NRANK,P)
      F3NORM=EPSI*PANORM*BNORM
      FNORM=DBLE(PNORM)*ANORM*F2NORM+DBLE(F1NORM)*BNORM+
     1 DBLE(F3NORM)
      C=GG*EPSI*NRANK*NRANK*(2.005+NRANK)
      TEST=1.D0-DBLE(FNORM)*PABNRM
      D=TEST**2-4.D0*DBLE(C)*PABNRM
      PABFNM=.5D0*(DBLE(TEST)-SQRT(D))/C
      F4NORM=DBLE(PABFNM)*(C*PABFNM+FNORM*PABNRM)
      ERROR=DBLE(PNORM)*(DBLE(BNORM)*F4NORM+DBLE(PABNRM)*
     1 (F2NORM+2*EPSI*BNORM))
      RELERR=ERROR/GNORM
      RETURN
      END
```

```
      SUBROUTINE NORM(X,M,N,A)
      DIMENSION A(50,50)
      DOUBLE PRECISION DP
C
C THIS SUBROUTINE COMPUTES THE INFINITY NORM OF THE M-BY-N
C MATRIX A AND STORES IT IN X.  INNER PRODUCTS ARE
C ACCUMULATED.
C
      X=0
      DO 10 I=1,M
      DP=0
      DO 5 J=1,N
 5    DP=DP+ABS(A(I,J))
      IF(DP-X) 10,10,6
 6    X=DP
 10   CONTINUE
      RETURN
      END
```

```
      FUNCTION AMAX(I,M,N,A)
      DIMENSION A(50,50)
C
C THIS FUNCTION SUBPROGRAM COMPUTES THE ELEMENT OF
C MAXIMUM MODULUS IN ROWS I,...,M AND COLUMNS I,...,N OF
C THE RECTANCULAR MATRIX A OF DIMENSION M-BY-N.
C
      AMAX=0
      DO 10 J=I,M
      DO 10 K=I,N
 10   AMAX=AMAX1(AMAX,ABS(A(J,K)))
      RETURN
      END
```